

Diego Cláudio da Silva

**MONITOR MULTIPARAMÉTRICO COM CONEXÃO WI-FI  
PARA USO PORTÁTIL E DOMICILIAR**

Trabalho de Conclusão de Curso  
submetido ao Departamento de  
Engenharia Elétrica e Eletrônica da  
Universidade Federal de Santa  
Catarina para a obtenção do título de  
Bacharel em Engenharia Eletrônica  
Orientador: Prof. Dr. Daniela Ota  
Hisayasu Suzuki.

Florianópolis - SC  
2018

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Silva, Diego Cláudio da  
Monitor Multiparamétrico com conexão Wi-Fi para  
uso portátil e domiciliar / Diego Cláudio da Silva  
; orientador, Daniela Ota Hisayasu Suzuki, 2018.  
130 p.

Trabalho de Conclusão de Curso (graduação) -  
Universidade Federal de Santa Catarina, Centro  
Tecnológico, Graduação em Engenharia Eletrônica,  
Florianópolis, 2018.

Inclui referências.

1. Engenharia Eletrônica. 2. Engenharia  
Biomédica. 3. Cuidados em saúde. 4. Monitor  
Multiparamétrico. 5. Internet das coisas. I.  
Suzuki, Daniela Ota Hisayasu. II. Universidade  
Federal de Santa Catarina. Graduação em Engenharia  
Eletrônica. III. Título.



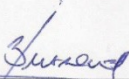


Diego Cláudio da Silva

**MONITOR MULTIPARAMÉTRICO COM CONEXÃO WI-FI PARA USO PORTÁTIL E DOMICILIAR**

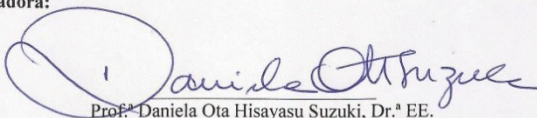
Este Trabalho foi julgado adequado para obtenção do Título de Bacharel em Engenharia Eletrônica e aprovado em sua forma final pela Banca Examinadora

Florianópolis, 15 de junho de 2018.

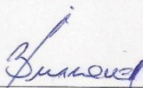


PhD. Jefferson Luiz Brum Marques, EE.  
Coordenador do Curso

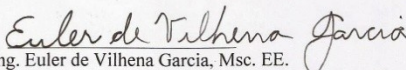
**Banca Examinadora:**



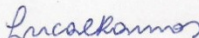
Prof.<sup>a</sup> Daniela Ota Hisayasu Suzuki, Dr.<sup>a</sup> EE.  
Orientadora  
Universidade Federal de Santa Catarina



PhD. Jefferson Luiz Brum Marques, EE.  
Universidade Federal de Santa Catarina



Dr. Eng. Euler de Vilhena Garcia, Msc. EE.  
Universidade de Brasília



Eng. Luciana Costa Ramos, Msc. EE.  
Universidade Federal de Santa Catarina



À memória de meu Avô Naldi e de  
meu primo Romário.





## AGRADECIMENTOS

Acima de tudo gostaria de agradecer aos meus pais, Cláudio Naldi da Silva e Marli Otilia Ventura da Silva, por serem os melhores pais com quem eu sequer poderia sonhar. Por me apoiarem em cada momento e decisão de minha vida. Mesmo com pouco estudo terem me dado a oportunidade de me formar em uma Universidade Federal. E principalmente, por terem dado a educação que me tornou quem eu tenho orgulho de ser.

Agradeço ainda:

Ao meu irmão mais velho, Maicon, que conviveu comigo durante toda a faculdade, dando conselhos fundamentais. Bem como meu irmão mais novo, Gabriel, que tornou meus finais de semana muito mais divertidos, alegrando os momentos difíceis de estudo.

À minha namorada, Priscila, por ser alguém com quem eu sempre pude contar, estando ao meu lado, dando suporte e sendo parceira, tornando meus últimos anos na UFSC muito mais agradáveis. Assim como meu sogro, José Ricardo, e minha sogra, Eliane, por me acolherem na família.

À professora Daniela, por ter me aceitado como bolsista logo no começo do curso, me ensinando e dando conselhos fundamentais ao longo da graduação. Além de todos os familiares, amigos e colegas, com quem pude contar, trabalhar e estudar durante todos esses anos.

Por fim gostaria de terminar com o trecho da música *“Learn to Fly”* da banda americana Foo Fighters: *“Make my way back home when I learn to fly high”*. Podendo dizer que após todas as alegrias e dificuldades que tive na graduação em Engenharia Eletrônica, eu posso voltar para casa sabendo que aprendi a voar alto.



*“When something is important enough, you do it  
even if the odds are not in your favor.”*  
(Elon Musk)



## RESUMO

Cuidado em saúde é um termo importante na sociedade, sendo um aspecto fundamental e objeto de pesquisa e desenvolvimento constante. Melhorar o acesso à recursos assistenciais, avaliações médicas e medições precisas são pontos fundamentais para a Engenharia Biomédica, que nos dias atuais estão cada vez mais ligados com assuntos como conectividade e acessibilidade. Buscar minimizar, tornar portátil, inteligente e quase que “invisível” são algumas das principais metas no desenvolvimento e aperfeiçoamento de tecnologias. Se tratando da área médica isto não é diferente. Este trabalho busca utilizar conceitos de Engenharia Eletrônica em conjunto com os aspectos listados, descrevendo passo a passo o processo de criação de um Monitor Multiparamétrico com conexão *Wi-Fi* para uso portátil e domiciliar. Cada etapa é abordada detalhadamente, começando pelo projeto dos circuitos de instrumentação para a saturação de oxigênio, pressão arterial e eletrocardiograma, alarme, conexão com a internet, configuração do display, circuitos de alimentação, desenho da placa de circuito impresso e *design* final do equipamento. Por ultimo é realizada uma avaliação do protótipo e validação dos resultados, comparando os dados coletados pelo equipamento projetado e por um monitor comercial, em um manequim robótico, com os sinais fornecidos por seu *software* controlador. Chegando a resultados satisfatórios, atingindo assim os objetivos propostos.

**Palavras-chave:** Cuidado em saúde, Engenharia Biomédica, Conectividade, Portátil, Inteligente, Monitor Multiparamétrico.



## ABSTRACT

Health care is an important term in society, being a fundamental aspect and object of constant research and development. Improving support for care resources, medical evaluations and accurate measurements are key points for Biomedical Engineering, which today are increasingly linked with issues such as connectivity and accessibility. Seeking to minimize, make portable, smart and almost “invisible” are some of the key goals in developing and perfecting Technologies. When it comes to the medical area this is no different. This work seeks to use concepts of Electronic Engineering in conjunction with the listed aspects, describing step by step the process of creating a Multiparametric Monitor with Wi-Fi connection for portable and home use. Each step is covered in detail, starting with the design of the instrumentation circuits for oxygen saturation, blood pressure and electrocardiogram, alarm, internet connection, display configuration, power circuits, printed circuit board design and equipment final design. Finally, a prototype evaluation and validation of results is performed comparing the data collected by the projected equipment and a commercial monitor on a robotic manikin with the signals provided by its controller software. Arriving at satisfactory results, thus reaching the proposed objectives.

**Keywords:** Health Care, Biomedical Engineering, Connectivity, Portable, Intelligent, Multiparametric Monitor.





## LISTA DE FIGURAS

Figura 1 – Monitor Multiparamétrico.....	34
Figura 2 – Conceitos básicos do ciclo PDCA. ....	34
Figura 3 – Comportamento da Hb com a incidência de luz. ....	40
Figura 4 – Diagrama de blocos do circuito completo de SpO2. ....	40
Figura 5 – Sensor de oximetria desenvolvido. ....	41
Figura 6 – Utilização do sensor de oximetria desenvolvido. ....	42
Figura 7 – Sensor de oximetria desenvolvido. ....	43
Figura 8 – Filtro passa-altas passivo com $f_c = 0,72$ Hz. ....	44
Figura 9 – Análise em frequência passa-altas passivo com $f_c = 0,72$ Hz. ....	44
Figura 10 – Análise de transiente passa-altas passivo com $f_c = 0,72$ Hz. ....	45
Figura 11 – Filtro passa-baixas ativo com $f_c = 2,34$ Hz e $g = 69$ .....	46
Figura 11 – Análise em frequência do filtro passa-baixas ativo com $f_c = 2,34$ Hz e $g = 69$ . ....	47
Figura 13 – Análise de transiente do filtro passa-baixas ativo com $f_c = 2,34$ Hz e $g = 69$ . ....	47
Figura 14 – Segunda etapa de filtragem. ....	48
Figura 15 - Buffer de saída. ....	49
Figura 16 – Circuito completo para aquisição da SpO2. ....	50
Figura 17 – Fluxograma de código para SpO2. ....	51
Figura 18 – Coifa e manguei de ar utilizados. ....	52
Figura 19 – Curva característica para a pressão arterial. ....	53
Figura 20 – Diagrama de blocos do circuito completo de pressão arterial. ....	54
Figura 21 – Micro bomba de ar DC. ....	55
Figura 22 – Circuito de controle da bomba de ar. ....	56
Figura 23 – Válvula solenóide de pressão. ....	57
Figura 24 – Circuito de controle da válvula solenóide. ....	57
Figura 25 – Sensor de pressão diferencial. ....	58
Figura 26 – Sensor de pressão diferencial. ....	59
Figura 27 – Amplificador diferencial com ganho de tensão de 340. ....	59
Figura 28 – Filtro passa-altas ativo com $f_c = 0,61$ Hz e ganho 3. ....	61
Figura 29 – Análise em frequência do filtro passa-altas ativo com $f_h = 0,61$ Hz e ganho 3. ....	61
Figura 30 – Análise de transiente do filtro passa-altas ativo com $f_h = 0,61$ Hz e ganho 3. ....	62
Figura 31 – Filtro passa-baixas ativo com $f_c = 53,12$ Hz e ganho 5.7...63	

Figura 32 – Análise em frequência do filtro passa-baixas ativo com $f_c = 53,12$ Hz e ganho 5,7. ....	63
Figura 33 – Análise de transiente do filtro passa-baixas ativo com $f_c = 53,12$ Hz e ganho 5,7. ....	64
Figura 34 – Circuito completo para aquisição não invasiva da pressão arterial. ....	65
Figura 35 – Fluxograma de código para a pressão arterial. ....	66
Figura 36 – Nódulo sinoatrial e propagação do sinal cardíaco.....	67
Figura 37 – Eletrodos e cabo utilizados para eletrocardiograma. ....	68
Figura38 – Pontos de contato dos eletrodos no corpo. ....	68
Figura 39 – Diagrama de blocos do circuito completo de eletrocardiograma. ....	69
Figura 40 – Circuito de terra virtual. ....	70
Figura 41 – Amplificador de instrumentação. ....	71
Figura 42 – Driver de perna direita. ....	72
Figura 43 – Filtro passa-altas ativo com $f_c = 0,61$ Hz e ganho 3. ....	72
Figura 44 – Filtro passa-baixas ativo com $f_c = 53,12$ Hz e ganho 5,7....	73
Figura 45 – Circuito completo para aquisição do eletrocardiograma. ....	74
Figura 46 – Fluxograma de código para o ECG. ....	75
Figura 47 – Circuito de alarme com Buzzer piezoelétrico. ....	76
Figura 48 – Fluxograma de código para o alarme. ....	76
Figura 49 – Módulo Wi-Fi ESP8266. ....	77
Figura 50 – Fluxograma de código para a conexão Wi-Fi. ....	79
Figura 51 – Página do banco de dados utilizado. ....	80
Figura 52 – Gráfico da pressão sistólica disponível na página. ....	81
Figura 53 – Programação por blocos para o aplicativo desenvolvido. ..	81
Figura 54 – Aplicativo desenvolvido. ....	82
Figura 55 – Display tátil de 5 polegadas utilizado. ....	83
Figura 56 – Shield regulador de tensão para o Display. ....	84
Figura 57 – Conexão entre Display e Arduino. ....	84
Figura 58 – Design gráfico e disposições dos botões no display.....	85
Figura 59 – Funcionalidades do Display. ....	86
Figura 60 – Bateria de Li-ion de 3,8 V 1900 mAh. ....	87
Figura 61 – Módulo carregador de bateria TP4046. ....	88
Figura 62 – Regulador de tensão ajustável MT3608 <i>Step Up</i> . ....	89
Figura 63 – Regulador de tensão de 7 V para 3V. ....	90
Figura 64 – Regulador de tensão de 7 V para 5V. ....	91
Figura 65 – Layout da placa de circuito impresso desenvolvida. ....	91
Figura 66 – Organização dos componentes na PCI. ....	92
Figura 67 – Placa desenvolvida. ....	93

Figura 68 – Caixa Patola / Case para Montagem 55 x 97 x 148 mm - PB-114. ....	94
Figura 69 – Montagem do equipamento na caixa. ....	94
Figura 70 – Disposição dos botões e conexões no monitor. ....	95
Figura 71– Disposição das conexões no monitor. ....	96
Figura 72 – Design frontal final do monitor projetado. ....	97



## LISTA DE TABELAS

Tabela 1 – Dados técnicos do Arduino Mega2560.....	36
Tabela 2 – Especificações da micro bomba de ar.....	55
Tabela 3 – Especificações da válvula solenóide. ....	57
Tabela 4 – Especificações do sensor de pressão diferencial. ....	58
Tabela 5 – Especificações do módulo ESP8266. ....	78
Tabela 6 – Especificações do Display. ....	83
Tabela 7 – Funcionalidade dos botões <i>touchscreen</i> do <i>display</i> .....	85
Tabela 8 – Especificações do módulo carregador de bateria TP4046...	88
Tabela 9 – Especificações do módulo regulador de tensão Step-Up....	89
Tabela 10 – Consumo e duração da bateria. ....	98
Tabela 11 – Tempo médio para obtenção das medidas. ....	98
Tabela 12 – Tempo médio para o envio via Wi-Fi dos dados.....	99
Tabela 13 – Custo individual dos componentes e total do produto.....	99
Tabela 14 – Dimensões finais e peso.....	100
Tabela 15 – Valores obtidos para BPM.....	102
Tabela 16 – Valores obtidos para a pressão arterial. ....	103
Tabela 17 – Valores obtidos para a saturação de oxigênio .....	103



## LISTA DE ABREVIATURAS E SIGLAS

ADC – *Analog to Digital Converter*  
AMPOP – Amplificador Operacional  
BPM – Batimentos Por Minuto  
CCS – Centro de Ciências da Saúde  
CI – Circuito Integrado  
CMRR – *Common Mode Rejection Ratio*  
DC – *Direct Current*  
DIA – Diastólica  
ECG – Eletrocardiograma  
EDA – *Eletronic Design Automation*  
Hb – Hemoglobina  
HL7 – *Health Level 7*  
I/O – *Input/Output*  
IEB – Instituto de Engenharia Biomédica  
INA – *Instrumentation Amplifier*  
IOT – *Internet Of Things*  
IP – *Internet Protocol adress*  
IR – Infravermelho  
LED – *Light Emitting Diode*  
LPS – Laboratório de Práticas Simuladas  
MAP – *Mean Arterial Pressure*  
MIT – *Massachusetts Institute of Tecnology*  
MMP – Monitor Multiparamétrico  
NiBP – *Non-Invasive Blood Pressure*  
PA – Pressão Arterial  
PCI – Placa de Circuito Impresso  
PDCA – *Plan, Do, Check, ACT*  
PPG – *Photoplethysmography*  
PWM – *Pulse Width Modulation*  
PQRST – Intervalos do sinal cardíaco  
RR – *Respiration Rate*  
RX – Receptor  
SpO2 – Saturação de oxigênio no sangue  
SYS – Sistólica  
SW – *Switch*  
TX – Transmissor  
UFSC – Universidade Federal de Santa Catarina  
USB – *Universal Serial Bus*









## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>31</b>
1.1	CONTEXTUALIZAÇÃO .....	31
1.2	OBJETIVOS .....	31
<b>1.2.1</b>	<b>Objetivo geral .....</b>	<b>31</b>
<b>1.2.2</b>	<b>Objetivos específicos .....</b>	<b>32</b>
1.3	ESTRUTURA DO DOCUMENTO .....	31
<b>2</b>	<b>CONCEITOS BÁSICOS .....</b>	<b>34</b>
2.1	MONITOR MULTIPARAMÉTRICO .....	34
2.2	INTERNET DAS COISAS .....	34
<b>2.2.1</b>	<b>Home Care .....</b>	<b>35</b>
<b>2.2.2</b>	<b>Banco de dados .....</b>	<b>35</b>
2.3	MICROCONTROLADOR.....	35
<b>2.3.1</b>	<b>Arduino Mega 2560.....</b>	<b>35</b>
2.4	CICLO PDCA .....	36
<b>3</b>	<b>PROJETO DO MONITOR MULTIPARAMÉTRICO ...</b>	<b>39</b>
3.1	SATURAÇÃO DE OXIGÊNIO.....	34
<b>3.1.1</b>	<b>Sensor projetado.....</b>	<b>41</b>
<b>3.1.2</b>	<b>Filtro passa-altas passivo .....</b>	<b>43</b>
<b>3.1.3</b>	<b>Filtro passa-baixas ativo .....</b>	<b>45</b>
<b>3.1.4</b>	<b>Segunda etapa de filtragem .....</b>	<b>48</b>
<b>3.1.5</b>	<b>Buffer de saída.....</b>	<b>48</b>
<b>3.1.6</b>	<b>Circuito completo de aquisição para SpO2.....</b>	<b>50</b>
<b>3.1.7</b>	<b>Código SpO2 .....</b>	<b>51</b>
3.2	PRESSÃO ARTERIAL.....	52
<b>3.2.1</b>	<b>Circuito de controle do motor DC .....</b>	<b>54</b>
<b>3.2.2</b>	<b>Circuito de controle da válvula solenóide .....</b>	<b>56</b>
<b>3.2.3</b>	<b>Sensor de pressão diferencial .....</b>	<b>58</b>

3.2.4	<b>Amplificador diferencial.....</b>	<b>59</b>
3.2.5	<b>Filtro passa-altas ativo.....</b>	<b>60</b>
3.2.6	<b>Filtro passa-baixas ativo.....</b>	<b>62</b>
3.2.7	<b>Circuito completo para aquisição da NiBP.....</b>	<b>65</b>
3.2.8	<b>Código para aquisição da NiBP .....</b>	<b>66</b>
3.3	<b>ELETROCARDIOGRAMA .....</b>	<b>67</b>
3.3.1	<b>Circuito de terra virtual .....</b>	<b>69</b>
3.3.2	<b>Amplificador de instrumentação .....</b>	<b>70</b>
3.3.3	<b>Circuito driver de perna direita .....</b>	<b>71</b>
3.3.4	<b>Filtro passa-altas .....</b>	<b>72</b>
3.3.5	<b>Filtro passa-baixas .....</b>	<b>73</b>
3.3.6	<b>Circuito completo para aquisição do ECG .....</b>	<b>74</b>
3.3.7	<b>Código para aquisição do sinal cardíaco e calculo BPM .</b>	<b>75</b>
3.4	<b>ALARME SONORO .....</b>	<b>75</b>
3.5	<b>CONECTIVIDADE COM A INTERNET.....</b>	<b>76</b>
3.5.1	<b>Módulo ESP8266.....</b>	<b>77</b>
3.5.2	<b>Banco de dados.....</b>	<b>79</b>
3.5.3	<b>Aplicativo.....</b>	<b>81</b>
3.6	<b>DISPLAY .....</b>	<b>82</b>
3.7	<b>ALIMENTAÇÃO DO SISTEMA .....</b>	<b>87</b>
3.7.1	<b>Bateria.....</b>	<b>87</b>
3.7.2	<b>Circuito de carga.....</b>	<b>87</b>
3.7.3	<b>Regulador de tensão <i>Boost Step Up</i> .....</b>	<b>88</b>
3.7.4	<b>Reguladores de tensão de 3 V e 5 V.....</b>	<b>89</b>
3.8	<b>DESENHO DA PLACA DE CIRCUITO IMPRESSO.....</b>	<b>91</b>
3.9	<b>DESIGN FINAL DO EQUIPAMENTO.....</b>	<b>93</b>
4	<b>AVALIAÇÃO DO PROTÓTIPO .....</b>	<b>97</b>
4.1	<b>AVALIAÇÃO DE CUSTOS E DESEMPENHO .....</b>	<b>97</b>

4.2	VALIDAÇÃO .....	100
5	CONSIDERAÇÕES FINAIS .....	105
	REFERENCIAS BIBLIOGRÁFICAS .....	109
	APENDICE A – Código Completo .....	115



# 1 INTRODUÇÃO

## 1.1 CONTEXTUALIZAÇÃO

No decorrer dos anos, os idosos tendem a perder sua autonomia física e com isso passam a ter maiores restrições em sua locomoção, o que pode dificultar ou até impossibilitar o seu acesso aos estabelecimentos de saúde. O mesmo pode ocorrer com pessoas com deficiências diversas, ou até mesmo crianças. Muitas doenças acabam exigindo cuidados preventivos, como avaliações diárias ou periódicas, que demandam do paciente a necessidade de locomoção até um local onde haja equipamentos adequados para o acompanhamento.

Com o avanço da tecnologia surgiram determinados equipamentos que possibilitam a medição de sinais vitais em casa, como aparelhos de pressão, oxímetros de pulso, dentre outros. Porém os aparelhos mais comuns no mercado têm funcionalidades limitadas, e aqueles mais completos, como monitores multiparamétricos, são caros e pouco acessíveis. Além disso esses equipamentos possibilitam apenas a medição, dando suporte ao diagnóstico dado por um profissional de saúde.

Para isso, uma solução seria o cuidado domiciliar, através da implementação de um monitor multiparamétrico sem fio, que possibilite medir de maneira simples o sinal cardíaco, a saturação de oxigênio e a pressão arterial. Assim, os dados medidos podem então ser enviados para um banco de dados via *wi-fi*, onde posteriormente ou até simultaneamente podem ser avaliados pelo médico ou instituição responsável.

Através do sistema de cuidado à distância, os custos podem diminuir possibilitando um atendimento mais abrangente. Pode-se citar como exemplo, o número de leitos hospitalares, que diminuiria substancialmente. Dessa maneira, todo esse processo, renderia maior qualidade de vida para os pacientes, especialmente os que são idosos ou com alguma deficiência, pois não teriam a necessidade de se locomover.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo geral

Projetar um dispositivo capaz de realizar a mensuração da pressão arterial, eletrocardiograma e oximetria de pulso, com uma *interface* simples, a qual o usuário consiga utilizar sem o auxílio técnico,

mostrando ainda as formas de onda e valores medidos em um display, além de disponibilizar os dados via *wi-fi* em um banco de dados *online*.

### 1.2.2 Objetivos específicos

Para atingir o objetivo geral, é necessária a contemplação dos seguintes objetivos específicos:

- Projetar um equipamento a custo abaixo dos produtos encontrados atualmente no mercado, com tamanho reduzido e bateria recarregável, possibilitando assim maior acessibilidade;
- Implementação de um display tátil que pode ser operado de forma amigável;
- Medição da pressão arterial, eletrocardiograma e oximetria;
- Implementação de alarme sonoro para o caso dos dados atingirem valores prejudiciais ao paciente;
- Conexão do equipamento via *Wi-Fi* para transmissão dos dados medidos do paciente;
- Desenvolvimento de aplicativo Android.

## 1.3 ESTRUTURA DO DOCUMENTO

Este projeto é dividido em seis capítulos. O capítulo 2 realiza uma revisão literária de conceitos chaves para o desenvolvimento do trabalho. O capítulo 3 consiste na descrição passo a passo de como o monitor multiparamétrico foi projetado. O capítulo 4 por sua vez aborda a implementação física do equipamento. Já o capítulo 5 demonstra como os testes finais foram feitos e expõe os resultados. Por último, no capítulo 6 é feita uma conclusão sobre tudo que foi abordado ao longo do projeto.





## 2 CONCEITOS BÁSICOS

### 2.1 MONITOR MULTIPARAMÉTRICO

Monitor multiparamétrico é um dos principais equipamentos médicos hospitalares utilizados por profissionais na área para visualizar e acompanhar indicadores de saúde do paciente. São dispositivos capazes de realizar medidas de parâmetros variados do corpo, exibindo em uma tela dados como eletrocardiograma (ECG), frequência cardíaca (BPM), saturação de oxigênio (SpO<sub>2</sub>), frequência respiratória (RR), temperatura (°C), pressão arterial não invasiva (NiBP), dentre outros.

É um equipamento de grande importância para hospitais e clínicas, pois permite analisar o estado de pacientes adultos, pediátricos, neonatos, e até possui modelos para uso veterinário. Outro aspecto importante é que não exige atendimento contínuo dos profissionais da saúde, pois após a conexão dos eletrodos e sensores necessários no paciente a detecção e exibição dos resultados são feitas de maneira contínua, contando ainda com um sistema de alarmes sonoros, que apitam caso algum dado monitorado fique fora dos limites estipulados.

Figura 1 – Monitor Multiparamétrico.



Fonte: BEMED. Retirado de [1].

## 2.2 INTERNET DAS COISAS

Internet das coisas, também chamada de IOT, do inglês *Internet of things* é um termo utilizado para descrever dispositivos que estão conectados à internet, captando, enviando e recebendo informações distintas de diversas maneiras. Estes equipamentos variam desde industriais até dispositivos pessoais, presentes no dia-a-dia das pessoas. Normalmente se comunicam com a Nuvem, enviando ou recebendo dados de sensores, comandos para atuadores, salvando informações em banco de dados, etc.

Atualmente há uma grande gama de aplicações para IOT, e na área de engenharia biomédica não é diferente. A internet das coisas nesta área normalmente está associada a conceitos como *home care*, *e-health*, *healthy home* e saúde ubíqua.

### 2.2.1 Home Care

*Home Care* é um termo em inglês utilizado para descrever cuidados domiciliares. Pode ser relacionado a cuidados que o paciente recebe em casa, sem a necessidade de se deslocar a um ambiente hospitalar, tais como cuidados médicos, nutricionais, fisioterápicos, etc.

### 2.2.2 Banco de dados

Banco de dados, ou base de dados, é um conjunto de informações relacionadas e armazenadas. É um conceito similar a uma biblioteca, porém digital. Atualmente existe uma variedade de serviços de banco de dados, sendo comuns serviços *online* para armazenamento de informações, também chamados de nuvem.

## 2.3 MICROCONTROLADOR

Um microcontrolador pode, de maneira simplificada, ser compreendido como um circuito integrado (CI) de uso geral, que tem a possibilidade de ser programado para executar ações específicas.

### 2.3.1 Arduino Mega 2560

O Arduino é uma plataforma de hardware livre para prototipagem de circuitos eletrônicos. Possui diversas versões, com tamanhos e configurações distintas. O Arduino Mega 2560 é atualmente um dos

modelos mais completos. Possuindo um microcontrolador ATmega2560, mais de 50 pinos de entrada e saída, e demais especificações conforme a Tabela 1. Permite a programação através de uma plataforma amigável, e tem uma grande base de informações disponíveis *online*.

Tabela 1 – Dados técnicos do Arduino Mega2560.

<b>Parâmetro</b>	<b>Resultado</b>
Microcontrolador	ATmega2560
Tensão de operação	5 V
Tensão de alimentação (recomendada)	7-12 V
Tensão de alimentação (limite)	6-20 V
Entradas e saídas digitais	54 das quais 15 podem ser PWM
Entradas analógicas	16
Corrente contínua por pino I/O	40 mA
Corrente contínua para o pino 3.3	50 mA
Memória Flash	256 KB dos quais 8 KB são usados pelo bootloader
Memória SRAM	8 KB
EEPROM	4 KB
Velocidade do Clock	16 MHz
Dimensões	101,6 mm x 53,4 mm
Peso	150 g

Fonte: Adaptado de [2].

## 2.4 CICLO PDCA

O ciclo PDCA é uma ferramenta de gestão, controle e planejamento. Suas siglas vem de Planejar (do inglês *Plan*), Fazer (do inglês *Do*), Checar (do inglês *Check*) e Agir (do inglês *Act*). Sua idéia consiste em uma sequência de ações e boas práticas a serem tomadas para a obtenção de um bom resultado em um projeto. Cada etapa pode ser melhor visualizada e compreendida através da Figura 2.

Figura 2 – Conceitos básicos do ciclo PDCA.



Fonte: Retirado de [3].



### 3 PROJETO DO MONITOR MULTIPARAMÉTRICO

O projeto foi realizado tendo como base os conceitos do ciclo PDCA. Como os sinais gerados pelo corpo atuam de maneiras distintas, diversas abordagens foram necessárias para captá-los. E entre os possíveis parâmetros fisiológicos que podem ser observados, os principais são: pressão arterial (PA), eletrocardiograma (ECG) e saturação de oxigênio (SpO<sub>2</sub>). Com a realização de uma pesquisa bibliográfica sobre o funcionamento desses parâmetros no corpo humano, foi possível utilizar os conhecimentos adquiridos para projetar e implementar circuitos capazes de fazer as aquisições.

Os esquemáticos, simulações, e placa de circuito impresso do projeto foram todos desenvolvidos na plataforma *online* livre *EASY EDA*. O processamento dos sinais aferidos e o controle das entradas e saídas foram feitos por um microcontrolador Atmega2560 responsável pelos cálculos dos parâmetros e por expor em um *display* as formas de onda e dados coletados, além de controlar um módulo *wi-fi* que irá disponibilizar os valores num banco de dados e em um aplicativo. Com exceção das características fisiológicas, que são descritas sem se aprofundar muito nos conceitos médicos, cada etapa é abordada detalhadamente nas seções a seguir.

#### 3.1 SATURAÇÃO DE OXIGÊNIO

O corpo possui uma proteína responsável pelo transporte de oxigênio pelo sangue, a hemoglobina (Hb). A hemoglobina oxigenada e a desoxigenada absorvem luz vermelha (660 nm de comprimento de onda) e infravermelha (900 nm de comprimento de onda) de maneiras diferentes. A Figura 3, ilustra o comportamento da Hb com a absorção de luz. É perceptível que o sinal de luz detectado possui amplitude e componente DC distintas para os dois comprimentos de onda.

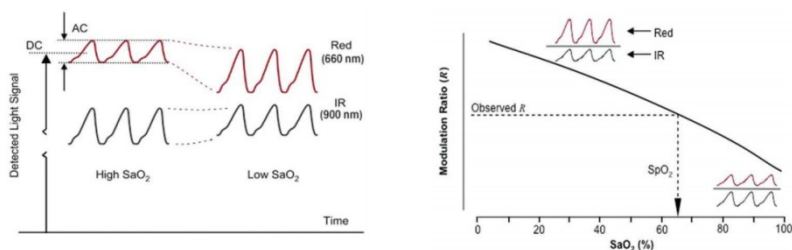
Realizando a razão entres os dois sinais, Equação (3.1), e manuseando o resultado, Equação (3.2), é possível chegar à uma reta aproximada (Figura 3), que relaciona a absorção de luz com a porcentagem de saturação de oxigênio no sangue (SpO<sub>2</sub>).

$$R = \frac{AC_{red} / DC_{red}}{AC_{ir} / DC_{ir}} \quad (3.1)$$

$$SpO_2 (\%) = 110 - 25R \quad (3.2)$$

A Equação (3.2) é frequentemente encontrada na literatura [4]. Ela é um modelo padrão para computar  $SpO_2$  (%) e sua demonstração não será feita aqui. Contudo seus valores precisam ser calibrados de maneira empírica para cada sensor.

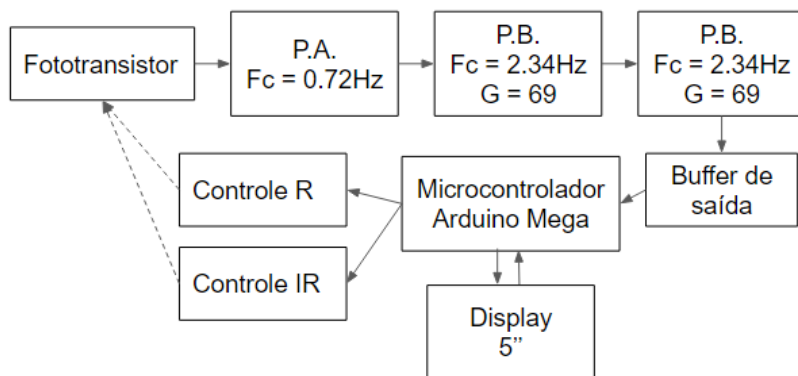
Figura 3 – Comportamento da Hb com a incidência de luz.



Fonte: Texas Instruments. Adaptado de [4].

A Figura 4 ilustra o diagrama de blocos completo do circuito de aquisição da onda plestimográfica (onda referente ao sinal de luz captado, conhecida como *photoplethysmography*/PPG) e da porcentagem de oxigênio. Cada etapa é abordada nas subseções a seguir.

Figura 4 – Diagrama de blocos do circuito completo de  $SpO_2$ .



Fonte: Elaboração própria.



### 3.1.1 Sensor projetado

Existem dois tipos mais comuns de sensores de PPG utilizados, os transmissivos e os reflexivos. O método transmissivo incide luz vermelha e infravermelha alternadamente em uma região do corpo que permita a captação da luz residual no outro lado, por exemplo, o dedo ou a orelha, e utiliza um fotodetector para averiguar a variação da luz transmitida. O método reflexivo funciona de maneira similar, porém captando a luz residual refletida.

Para este projeto foi desenvolvido um sensor reflexivo (Figura 5), utilizando LEDs vermelho, infravermelho e um fototransistor, com a aquisição do sinal realizada no dedo (Figura 6) e com um custo reduzido em relação aos sensores comerciais.

Figura 5 – Sensor de oximetria desenvolvido.



Fonte: Elaboração própria.

Figura 6 – Utilização do sensor de oximetria desenvolvido.

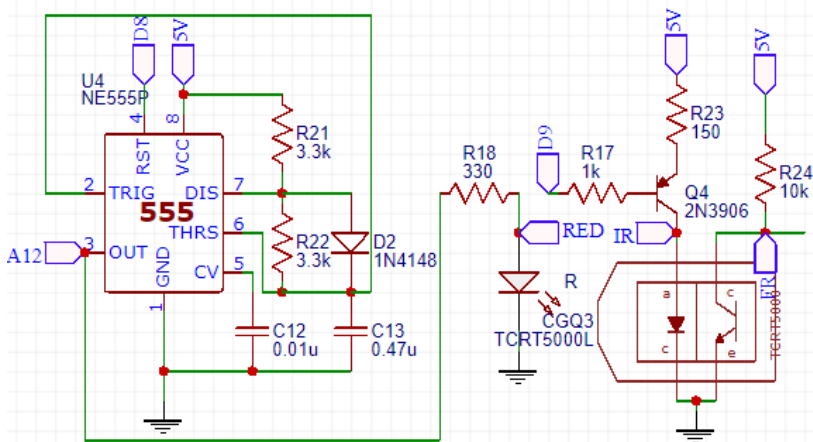


Fonte: Elaboração própria.

Como o cálculo da  $SpO_2$  (%) é realizado através da razão entre os sinais captados referentes à luz vermelha e à luz infravermelha, é necessário um controle dos LEDs do sensor, assim o circuito da Figura 7 foi desenvolvido.

Inicialmente temos um CI NE555 atuando como oscilador astável, com *duty cycle* de quase 50%, e frequência de 464,21 Hz. Quando a saída do 555 está em um valor alto, o LED vermelho ascende, e o microcontrolador percebe que a entrada analógica A12 está em 5 V, configurando assim a saída digital D9 em um valor lógico baixo, desligando o emissor IR. Quando a saída do 555 está em 0 V o processo inverso ocorre.

Figura 7 – Sensor de oximetria desenvolvido.



Fonte: Elaboração própria.

### 3.1.2 Filtro passa-altas passivo

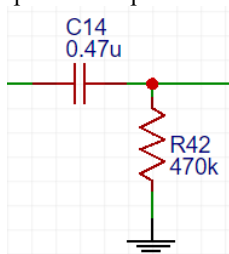
O sinal na saída do sensor possui baixa amplitude e está sujeito à ruído, portanto é necessário a utilização de uma instrumentação para filtrar e amplificar o sinal de maneira adequada. Assim o primeiro passo é o uso de um filtro passa-altas (Figura 8), com o intuito de remover a componente DC do sinal. Para tanto optou-se por implementar um filtro RC passivo, com frequência de corte de teórica de 0,72 Hz e real de 0,68 Hz. As Equações 3.3, 3.4 e 3.5 caracterizam a função de transferência e a frequência de corte do circuito. Nas figuras 9 e 10 é possível visualizar a análise em frequência e ao transiente.

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{s}{s + \frac{1}{RC}} \quad (3.3)$$

$$G(s) = \frac{s}{s + 4.53} \quad (3.4)$$

$$fc = \frac{1}{2\pi RC} = 0.72 \text{ Hz} \quad (3.5)$$

Figura 8 – Filtro passa-altas passivo com  $f_c = 0,72$  Hz.

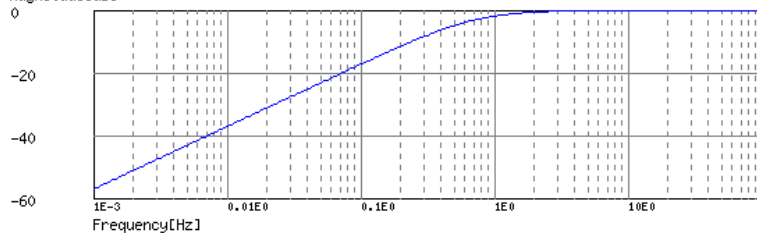


Fonte: Elaboração própria.

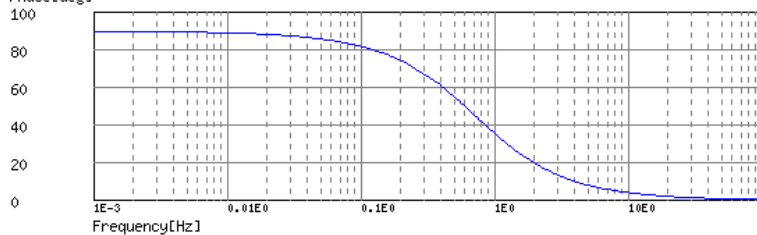
Figura 9 – Análise em frequência passa-altas passivo com  $f_c = 0,72$  Hz.

**BodeDiagram**

Magnitude[dB]

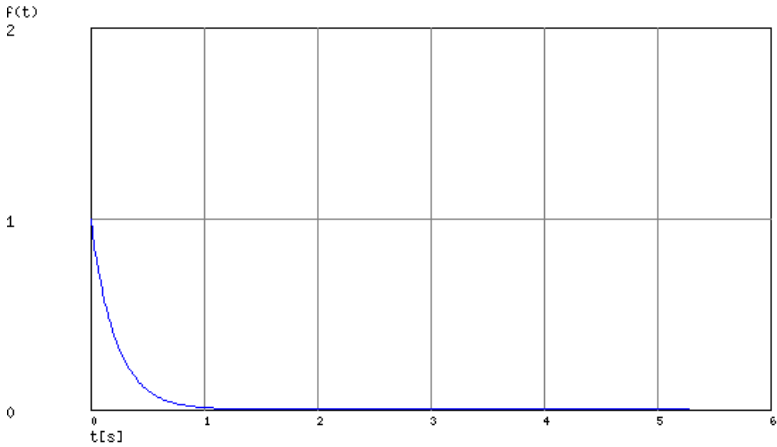


Phase[deg]



Fonte: Adaptado de [5].

Figura 10 – Análise de transiente passa-altas passivo com  $f_c = 0,72$  Hz.  
StepResponse



Fonte: Adaptado de [5].

### 3.1.3 Filtro passa-baixas ativo

Após a componente DC ser removida do sinal, ele ainda possui uma baixa amplitude para ser utilizado pelo microcontrolador, e está sujeito ao ruído de 60 Hz da rede elétrica, por isso um estágio composto de um filtro passa-baixas ativo foi desenvolvido (Figura 11), com o amplificador operacional operando em modo não inversor e intuito de amplificar o sinal com um ganho teórico de 69 V/V (real de 70,35) e uma frequência de corte teórica de 2,34 Hz (Real de 2,33 Hz), como mostra as equações abaixo.

$$G(s) = \frac{s+1014.71}{s+14.71} \quad (3.6)$$

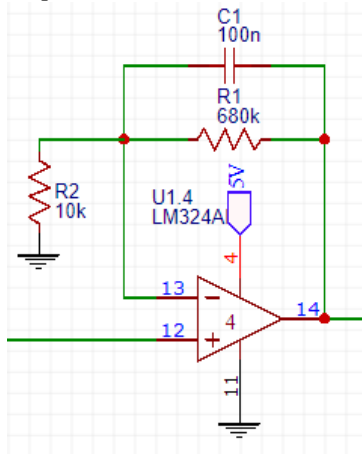
$$|p| = 2.34 \text{ Hz} \quad (3.7)$$

$$g(\infty) = 69 \quad (3.8)$$

A frequência de corte utilizada foi escolhida com base na frequência cardíaca, estipulando um pior caso onde ocorram 140 BPM,

tendo em vista que o fluxo sanguíneo e conseqüentemente o sinal do PPG variam com os batimentos cardíacos.

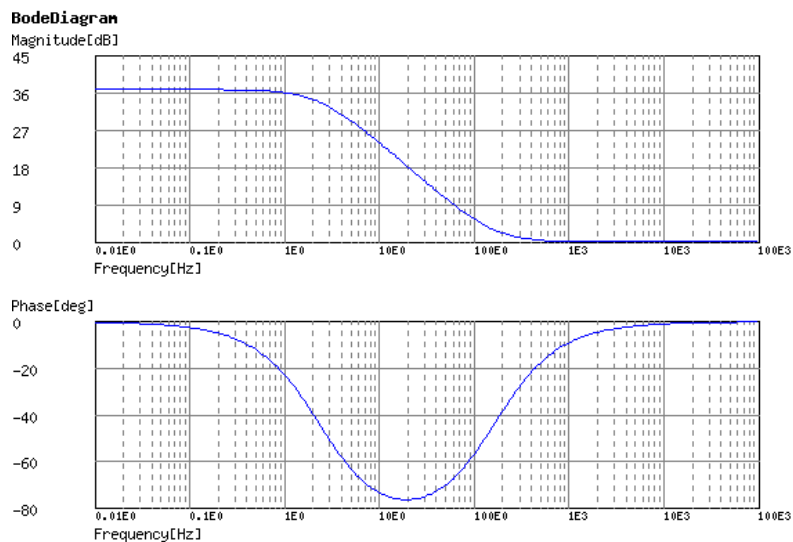
Figura 11 – Filtro passa-baixas ativo com  $f_c = 2,34 \text{ Hz}$  e  $G = 69$ .



Fonte: Elaboração própria.

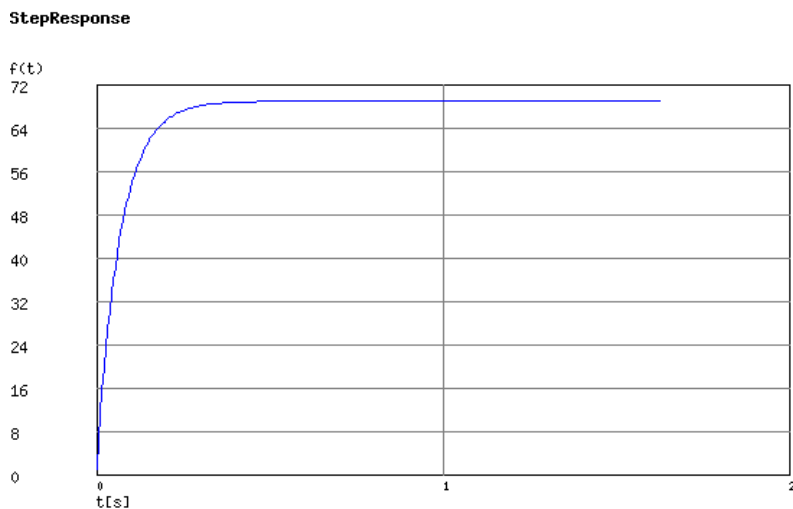
Nas Figuras 12 e 13 é possível visualizar a análise em frequência e ao transiente do filtro projetado.

Figura 12 – Análise em frequência do filtro passa-baixas ativo com  $f_c = 2,34 \text{ Hz}$  e  $G = 69$ .



Fonte: Adaptado de [6].

Figura 13 – Análise de transiente do filtro passa-baixas ativo com  $f_c = 2,34 \text{ Hz}$  e  $G = 69$ .

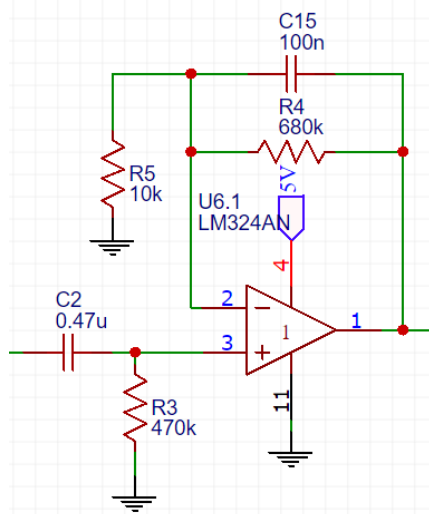


Fonte: Adaptado de [6].

### 3.1.4 Segunda etapa de filtragem

Uma segunda etapa de filtragem (Figura 14) foi implementada com configuração igual ao filtro ativo passa-altas do primeiro estágio e do filtro passa baixas ativo, cujo objetivo é aumentar a ordem do filtro e o ganho, melhorando assim a qualidade do sinal de saída. Com os dois amplificadores o ganho total do circuito é  $69 \times 69 = 4761 \text{ V/V}$ .

Figura 14 – Segunda etapa de filtragem.



Fonte: Elaboração própria.

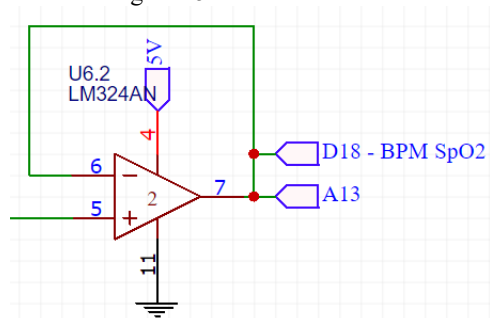
### 3.1.5 Buffer de saída

O último estágio desenvolvido foi o de um buffer não-inversor de ganho unitário (Figura 15), para isolar e conectar o restante do circuito ao microcontrolador, já que o buffer permite conectar um estágio de alta impedância de entrada a uma baixa carga de saída, diminuindo assim a impedância de saída do circuito e adequando ele para ser conectado à entrada analógica/ conversor ADC da placa.

$$V_{out} = V_{in} \quad (3.9)$$



Figura 15 - Buffer de saída.

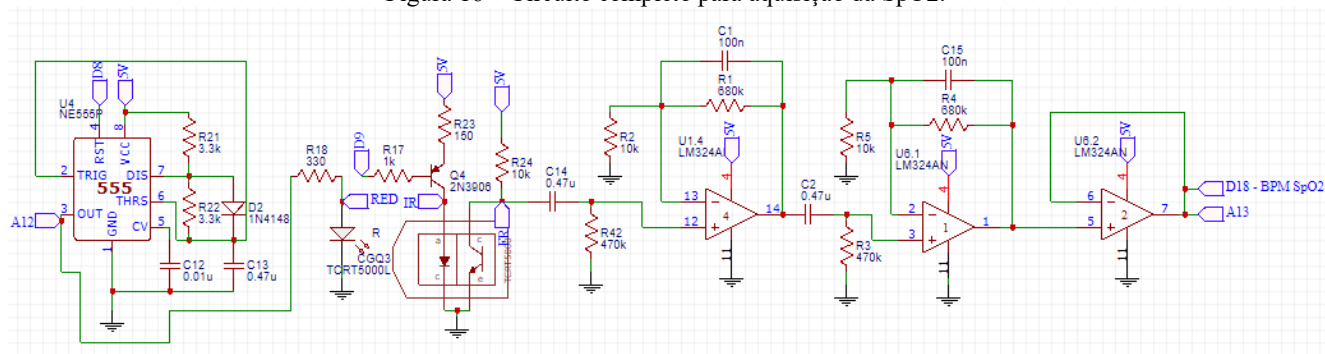


Fonte: Elaboração própria.

### 3.1.6 Circuito completo de aquisição para SpO2

Através de todas as etapas descritas nas seções anteriores foi possível, projetar, simular e testar o circuito completo para a aquisição da onda do PPG e para o cálculo de saturação de oxigênio no sangue. Obtendo-se assim o circuito completo da Figura 16.

Figura 16 – Circuito completo para aquisição da SpO2.



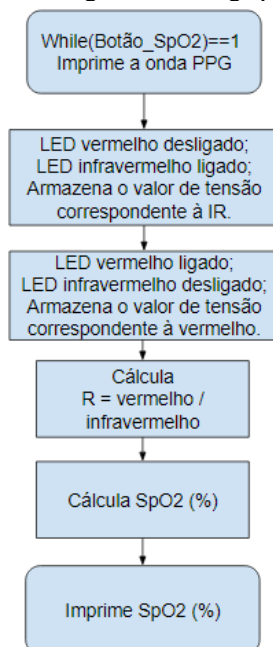
Fonte: Elaboração própria.

### 3.1.7 Código SpO2

Com o circuito pronto o próximo passo foi o desenvolvimento do código para captar, tratar e imprimir a o PPG e SpO2 (%). O algoritmo desenvolvido segue o fluxograma de código da Figura 17 e o código completo pode ser visto no Apêndice A.

Quando o botão correspondente a oximetria é ativado a saída do microcontrolador D8 é ligada, e consequentemente o 555 começa a oscilar. Paralelo a isso a entrada analógica A12 é analisada, captando através de A13 o valor correspondente à onda com absorção de luz vermelha quando a saída do 555 estiver em valor alto, e captando o valor correspondente à onda infravermelha quando o 555 estiver em valor lógico baixo. Em seguida o cálculo da razão entre as amplitudes das ondas é feito, e então utiliza-se a Equação (3.2) para obter SpO2 (%). Por ultimo o valor da porcentagem e a forma de onda PPG são impressos no display. Vale ressaltar que a Equação (3.2) possui valores distintos na prática, obtidos de maneira empírica através de uma série de testes e calibração, como pode ser visto no Apêndice A.

Figura 17 – Fluxograma de código para SpO2.

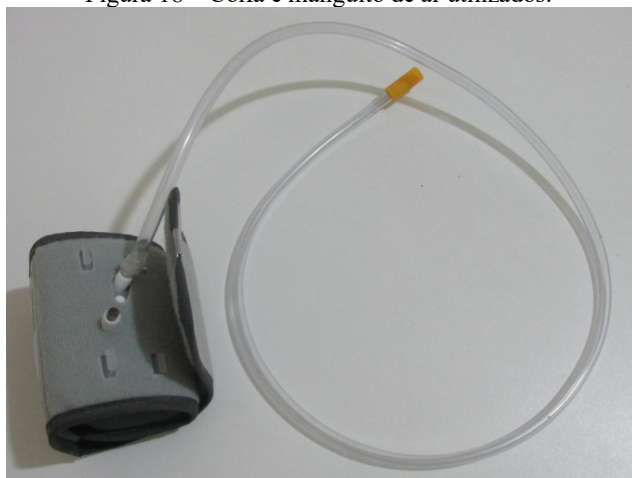


Fonte: Elaboração própria.

### 3.2 PRESSÃO ARTERIAL

A pressão arterial é a pressão que o sangue exerce contra a parede da artéria. Ela varia de acordo com os movimentos de contração e relaxamento do coração. Sendo que a pressão máxima ocorre quando o coração se contrai e bombeia sangue, e é conhecida como pressão sistólica. A pressão mínima ocorre no momento de relaxamento do coração, quando ele se infla de sangue, e é conhecida como pressão diastólica. A pressão arterial é descrita pelas suas pressões máxima (Sistólica – SYS) e mínima (diastólica – DIA) em mmHg (milímetros de mercúrio), sendo seu valor ideal 120/80 mmHg, e pode ser utilizada para o diagnóstico de doenças como a hipertensão. A averiguação da pressão pode ser feita inflando e desinflando um manguito (também chamado de coifa de ar, Figura 18) conectado em volta do braço ou do pulso, e aferindo a pressão através de um sensor, coletando assim os valores da pressão média, máxima e mínima na medida em que a pressão exercida pelo ar aumenta e diminui.

Figura 18 – Coifa e manguito de ar utilizados.



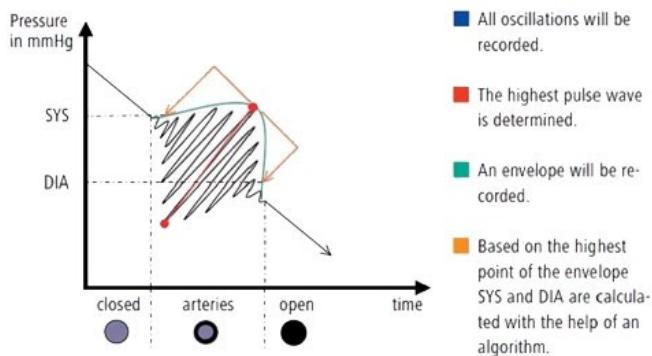
Fonte: Elaboração própria.

Há métodos distintos para a obtenção da pressão arterial. Neste projeto foi utilizado o método não invasivo (NiBP) oscilométrico. No método oscilométrico a coifa de ar no pulso do usuário é inflada até que possua pressão o suficiente para pressionar bem o pulso. Em seguida ela é lentamente esvaziada, e enquanto isso se analisa, com o auxílio de um

sensor de pressão, as oscilações da pressão na coifa. As oscilações seguem o padrão da Figura 19, inicialmente o pulso está bem pressionado e quase não é possível observar as oscilações, mas conforme o ar é expelido as oscilações aumentam de amplitude até atingir o valor máximo, que corresponde a pressão arterial média (MAP), após isso a amplitude das oscilações começa a diminuir.

A pressão MAP corresponde a pressão na coifa no momento que a maior oscilação ocorre, e com ela é possível estimar as pressões diastólica e sistólica. Para o projeto foram utilizadas equações simples, com valores obtidos de maneira empírica, estimando DIA como sendo  $0,86 * MAP$  e SYS como  $1,29 * MAP$ .

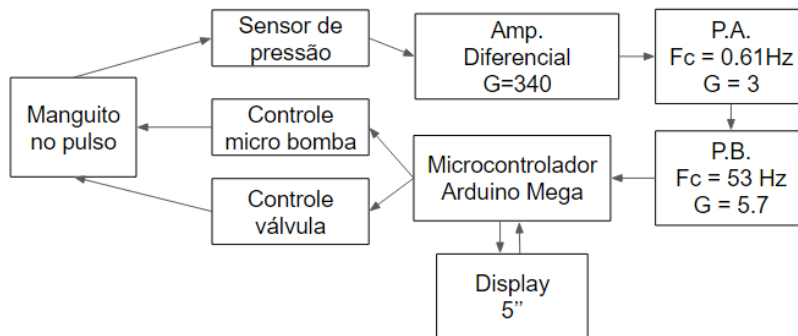
Figura 19 – Curva característica para a pressão arterial.



Fonte: Retirado de [7].

O funcionamento do circuito completo para obtenção da NiBP segue o diagrama de blocos abaixo, e cada etapa é descrita detalhadamente nas subseções a seguir.

Figura 20 – Diagrama de blocos do circuito completo de pressão arterial.



Fonte: Elaboração própria.

### 3.2.1 Circuito de controle do motor DC

Com a finalidade de encher o manguito de ar de maneira automática uma micro bomba de ar DC (Figura 21) é utilizada, operando com uma tensão DC de 3 V e corrente máxima de 150 mA, conforme as especificações da tabela abaixo. Tendo em vista que a tensão de operação dos pinos I/O do Arduino Mega 2560 é de 5V, que sua corrente máxima suportada é de 40 mA, e que a corrente máxima no pino 3,3 V é 50 mA, foi necessário desenvolver um circuito de controle para a ativação da bomba (Figura 22).

O circuito é constituído de um CI optoacoplador 4N25M, que opera através de um par LED/fototransistor. Quando o pino digital D11 do arduino é setado em *HIGH* o fluxo de corrente passa pelo LED do CI, que por sua vez ilumina o fototransistor permitindo assim o fluxo de corrente entre o coletor e o emissor, desta forma ativando a micro bomba de ar. Tendo em vista que sua tensão de operação é de 3 V, e que a tensão de alimentação do arduino é 7V, o terminal positivo do motor é conectado ao regulador de tensão de 3V (seção 4.1.4).

Tabela 2 – Especificações da micro bomba de ar.

<b>Tensão recomendada</b>	3 V
<b>Range de operação</b>	2 V ~ 3.2 V
<b>Tensão de entrada</b>	< 150 mA
<b>Tempo para inflar</b>	< 8s (de 0 à 300 mmHg)
<b>Fluido aplicado</b>	Ar
<b>Fluxo de ar</b>	> 70 ml/min
<b>Pressão máxima</b>	> 360 mmHg
<b>Dispersão</b>	Max 0.6 mmHg/ 15 seg a partir de 300 mmHg
<b>Tensão para religar</b>	2V
<b>Ruído</b>	62 dB (30 cm de distância)
<b>Vida útil</b>	30000 vezes

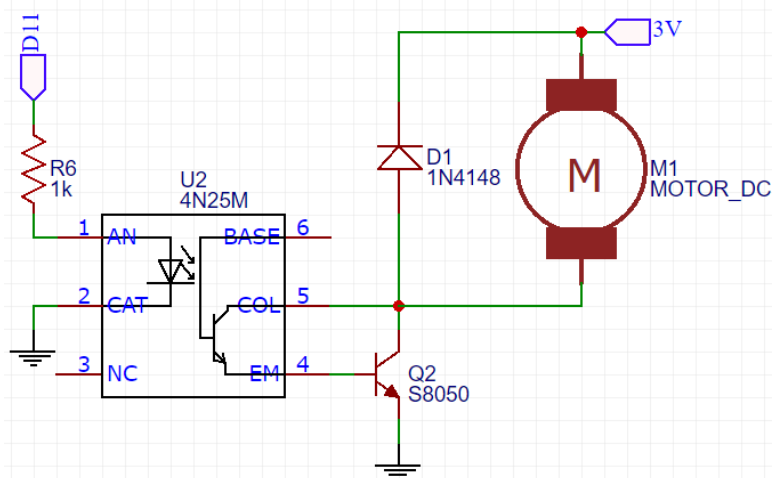
Fonte: Adaptado de[8].

Figura 21 – Micro bomba de ar DC.



Fonte: Retirado de [8].

Figura 22 – Circuito de controle da bomba de ar.



Fonte: Elaboração própria.

### 3.2.2 Circuito de controle da válvula solenóide

De maneira similar à micro bomba de ar, uma válvula solenóide de 3 V (Figura 23) é necessária, com o intuito de esvaziar o ar de maneira controlada, avaliando a pressão conforme o ar flui.

Assim como a bomba, a válvula também é alimentada com 3 V (especificações na Tabela 3), porém, como sua corrente máxima é menor, um circuito mais simples (Figura 24) é utilizado para seu acionamento, que é feito através da saída digital D12 do Arduino. A válvula tem seu estado normal como aberto, assim sendo quando D12 é acionado com um valor alto (5 V), a base do transistor é polarizada, permitindo o fluxo de corrente que percorre a válvula solenóide, travando e impedindo a saída de ar. O controle do fluxo de ar é feito com uma modulação de pulso (PWM), abrindo e fechando a válvula e assim esvaziando aos poucos o manguito.



Tabela 3 – Especificações da válvula solenóide.

<b>Tensão considerada</b>	3 V
<b>Corrente considerada</b>	75 mA
<b>Resistência DC</b>	40 ohms
<b>Consumo de potência</b>	0.225 W
<b>Velocidade de exaustão 300mmHg – 15 mmHg</b>	< 3 s
<b>Vazão</b>	< 3 mmHg/min
<b>Tensão mínima de operação</b>	< 2,3 V
<b>Tipo</b>	Normalmente aberta

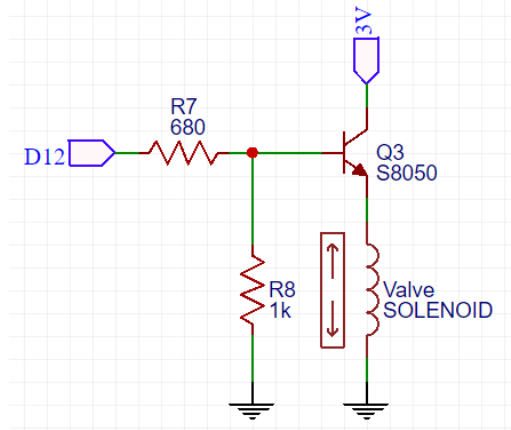
Fonte: Adaptado de [9].

Figura 23 – Válvula solenóide de pressão.



Fonte: Retirado de [9].

Figura 24 – Circuito de controle da válvula solenóide.



Fonte: Elaboração própria.

3.2.3 Sensor de pressão diferencial

Para medir a pressão arterial através da pressão na coifa de ar, é utilizado um sensor de pressão diferencial MPS20N0040D-D (especificações na Tabela 4), o qual possui duas entradas de ar, uma externa por onde o ar ambiente entra, e uma entrada que é conectada ao manguito. O sensor atua como uma ponte formada por quatro resistores de 5 kOhms, e um desses resistores varia sua resistência de acordo com a diferença de pressão entre as entradas de ar. A variação da resistência com a pressão ocorre de maneira linear. Assim, alimentando o sensor com 5 V, tem-se uma tensão diferencial  $V_2 - V_1$  (Figura 26) que depende da pressão. Desta forma é possível calibrar o sensor e estipular uma fórmula para a pressão em relação à tensão nas saídas do sensor.

Figura 25 – Sensor de pressão diferencial.



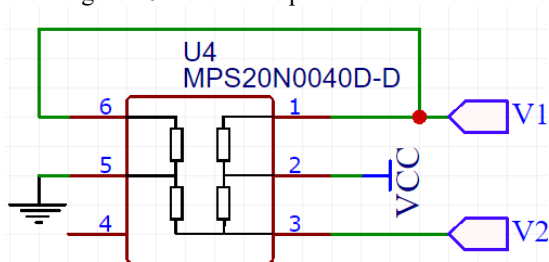
Fonte: Retirado de [10].

Tabela 4 – Especificações do sensor de pressão diferencial.

<b>Faixa de pressão medida</b>	300 mmHg / 5.8 PSI / 40 KPa
<b>Capacidade máxima de pressão</b>	3 vezes a faixa
<b>Tensão de operação</b>	5 V
<b>Impedância de entrada</b>	4 – 6 Kohms
<b>Impedância de saída</b>	4 – 6 Kohms
<b>Temperatura de operação</b>	-40 – -125 °C
<b>Meio acessível</b>	Limpo, seco e gases não corrosivos
<b>Variação de tensão</b>	+/- 25 mV
<b>Escala completa de tensão de saída</b>	50 – 100 mV
<b>Resistência de ponte</b>	4 – 6 Kohms

Fonte: Adaptado de [11].

Figura 26 – Sensor de pressão diferencial.

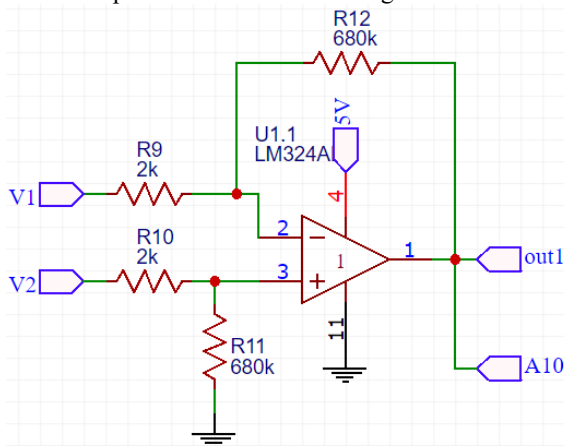


Fonte: Elaboração própria.

### 3.2.4 Amplificador diferencial

Como a tensão diferencial nas saídas do sensor é muito pequena (0~14 mV nos testes) para ser conectada direto ao microcontrolador, é necessário realizar uma etapa de ganho, para isso uma amplificador diferencial (Figura 27) é implementado, com ganho de 340 V/V, dado pela equação (3.9).

Figura 27 – Amplificador diferencial com ganho de tensão de 340.



Fonte: Elaboração própria.

Para  $R_9 = R_{10}$  e  $R_{11} = R_{12}$ , temos:

$$V_o = \frac{R_{11}}{R_9} * [V_2 - V_1] \quad (3.9)$$

A saída do amplificador diferencial é conectada ao restante do circuito e ao pino A10 do arduino, para que assim a pressão da coifa seja monitorada enquanto está sendo inflada e para que a válvula solenóide seja aberta ou fechada de acordo com a pressão atingida na coifa.

### 3.2.5 Filtro passa-altas ativo

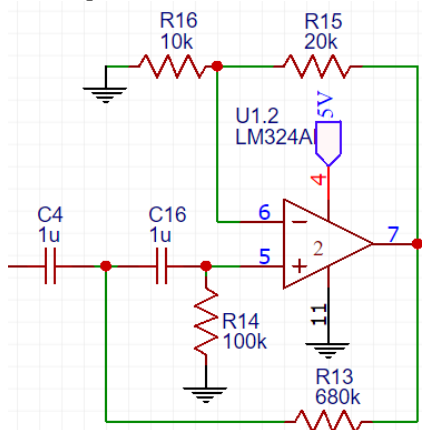
Após a saída do amplificador o sinal varia linearmente com o aumento da pressão de 0 à 5 V, sendo que as oscilações referentes as variações cardíacas possuem amplitude máxima de 10 mV em torno do valor linear. Assim é necessária a utilização de um filtro passa-altas ativo (Figura 28) de ganho relativamente baixo para eliminar a componente DC do sinal obtendo-se assim uma maior resolução das oscilações sem ocorrer saturação. Desta maneira um filtro Sallen-Key foi implementado com frequência de corte teórica de 0.61 Hz (real de 0,60 Hz) e ganho teórico de 3 V/V (real de 3,01) conforme as especificações abaixo.

$$\frac{V_o}{V_i} = \frac{Gs^2}{s^2 + s\frac{1}{C_1R_2} + s\frac{1}{C_2R_2} + s\frac{1}{C_1R_1}(1-G) + \frac{1}{C_1C_2R_1R_2}} \quad (3.10)$$

$$G = \frac{R_3 + R_4}{R_3} = 3 \text{ V/V} \quad (3.11)$$

$$fc = \frac{1}{2\pi\sqrt{R_1C_1R_2C_2}} = 0.61 \text{ Hz} \quad (3.12)$$

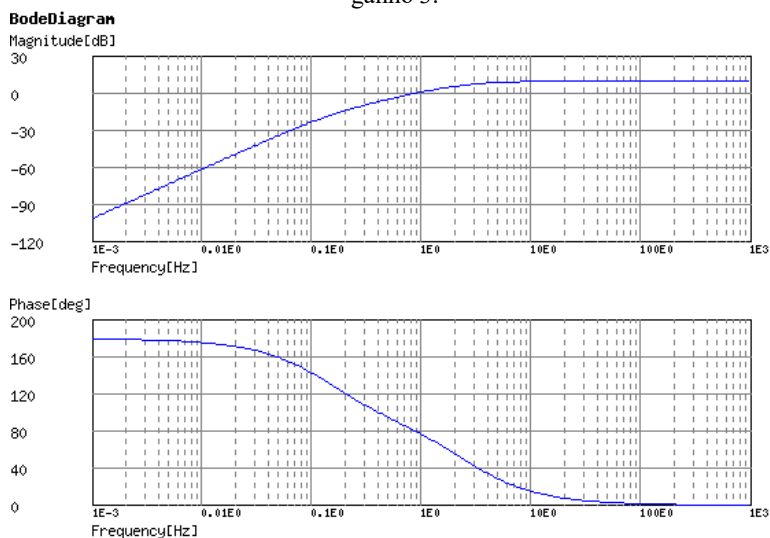
Figura 28 – Filtro passa-altas ativo com  $f_c = 0,61$  Hz e ganho 3.



Fonte: Elaboração própria.

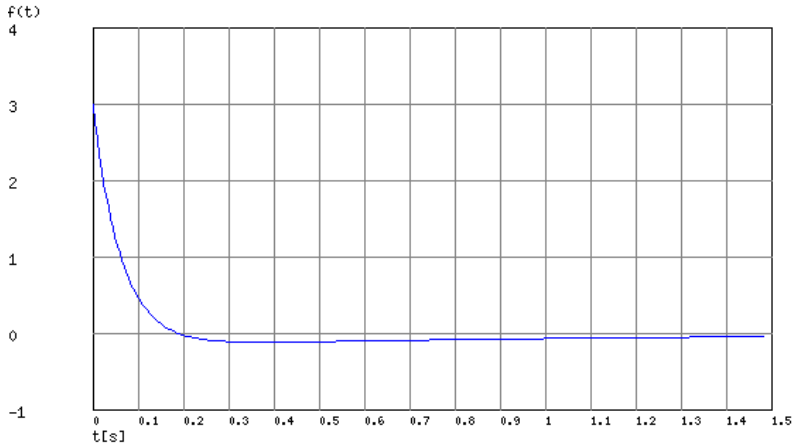
Nas Figuras 29 e 30 é possível ver o comportamento do filtro projetado, com amplitude e fase na análise em frequência, bem como sua análise de transiente.

Figura 29 – Análise em frequência do filtro passa-altas ativo com  $f_c = 0,61$  Hz e ganho 3.



Fonte: Adaptado de [12].

Figura 30 – Análise de transiente do filtro passa-altas ativo com  $f_c = 0,61$  Hz e ganho 3.



Fonte: Adaptado de 12.

### 3.2.6 Filtro passa-baixas ativo

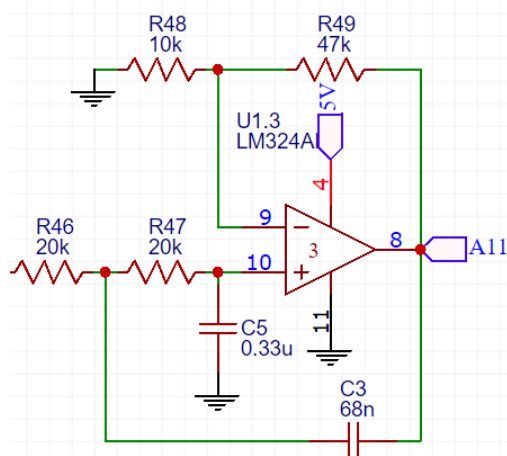
Para eliminar a influência de ruído da rede e obter um sinal o mais limpo possível, foi implementado um filtro passa-baixas ativo Sallen-Key (Figura 31), com frequência de corte teórica de 53,12 Hz (real de 56,35 Hz) e ganho teórico de 5,7 V/V (real de 5,77), como pode ser visto nas equações e gráficos abaixo.

$$\frac{V_o}{V_i} = \frac{\frac{G}{C_1 C_2 R_1 R_2}}{s^2 + s \frac{1}{C_1 R_2} + s \frac{1}{C_2 R_1} + s \frac{1}{C_2 R_2} (1-G) + \frac{1}{C_1 C_2 R_1 R_2}} \quad (3.13)$$

$$G = \frac{R_3 + R_4}{R_3} = 5,7 \text{ V/V} \quad (3.14)$$

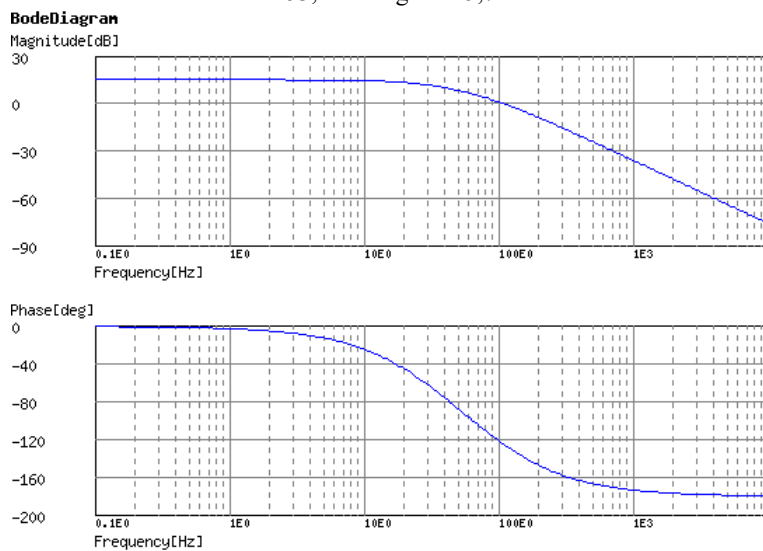
$$f_c = \frac{1}{2\pi\sqrt{R_1 C_1 R_2 C_2}} = 53,12 \text{ Hz} \quad (3.15)$$

Figura 31 – Filtro passa-baixas ativo com  $f_c = 53,12 \text{ Hz}$  e ganho 5,7.



Fonte: Elaboração própria.

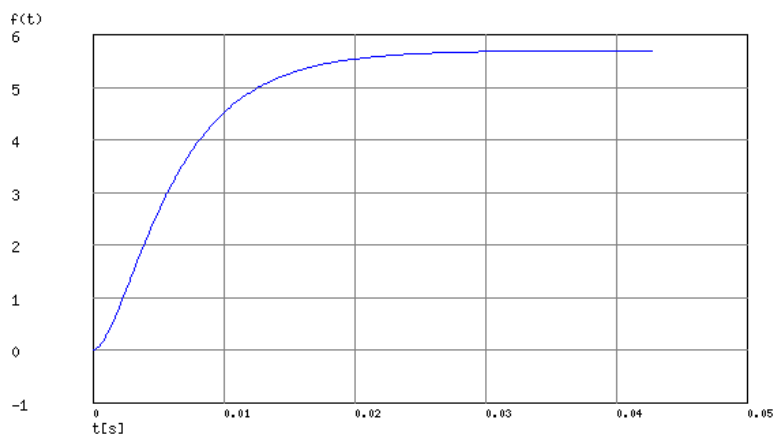
Figura 32 – Análise em frequência do filtro passa-baixas ativo com  $f_c = 53,12 \text{ Hz}$  e ganho 5,7.



Fonte: Adaptado de [12].

Figura 33 – Análise de transiente do filtro passa-baixas ativo com  $f_c = 53,12$  Hz e ganho 5,7.

**StepResponse**



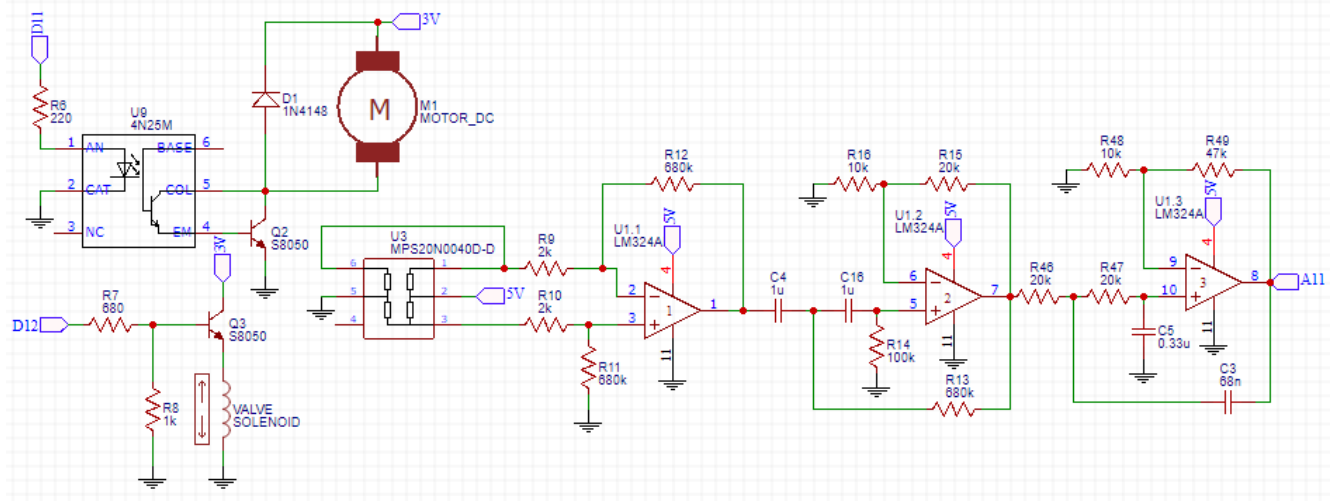
Fonte: Adaptado de [12].



### 3.2.7 Circuito completo para aquisição da NiBP

Com o projeto, simulação e testes, foi possível chegar ao circuito completo da figura abaixo para a aquisição não invasiva da pressão arterial.

Figura 34 – Circuito completo para aquisição não invasiva da pressão arterial.



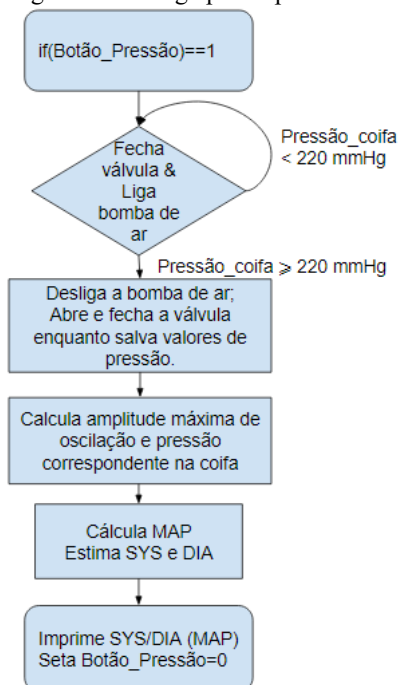
Fonte: Elaboração própria.

### 3.2.8 Código para aquisição da NiBP

A última etapa para a obtenção da pressão arterial é realizada pelo microcontrolador. Toda programação está disponível no Apêndice A, porém para melhor compreensão o fluxograma de código abaixo foi desenvolvido.

Assim que o botão relativo à pressão é pressionado, os pinos D12 e D11 são setados em *HIGH* para fechar a válvula e para que a bomba comece a encher o manguito. Então aquisições da entrada A10 são feitas visando saber se a pressão máxima desejada na coifa foi atingida, a qual normalmente ocorre em cerca de 6 à 10 segundos. Em seguida a bomba de ar é desligada e a válvula é controlada por uma modulação PWM escoando lentamente o ar. Enquanto isso aquisições de A10 e A11 são feitas, a fim de saber se a pressão na coifa correspondente a maior oscilação. Com esses dados, e após esvaziar o manguito, é possível então obter a pressão média e estimar a pressão sistólica e diastólica. Por ultimo os valores são impressos no display e o botão é setado em 0.

Figura 35 – Fluxograma de código para a pressão arterial.

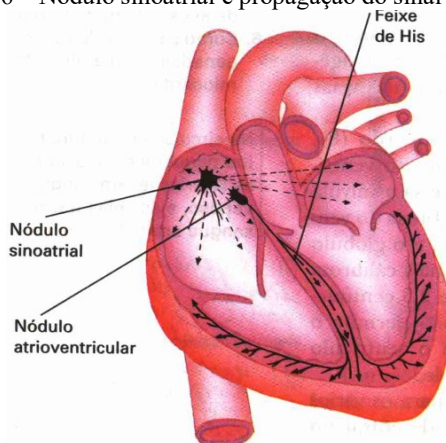


Fonte: Elaboração própria.

### 3.3 ELETROCARDIOGRAMA

No coração, no nódulo Sinoatrial (Figura 36) surge um potencial elétrico que se propaga pelas células cardíacas coordenando os movimentos de contração e relaxamento do coração. O eletrocardiograma consiste na mensuração de tais oscilações elétricas.

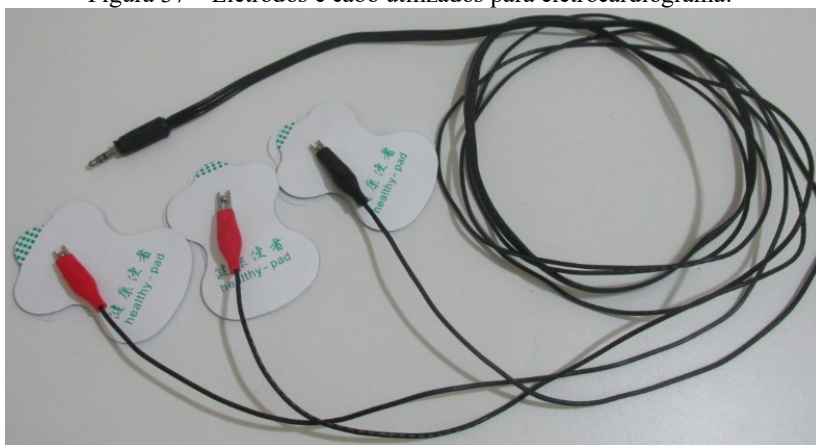
Figura 36 – Nódulo sinoatrial e propagação do sinal cardíaco.



Fonte: Adaptado de [13].

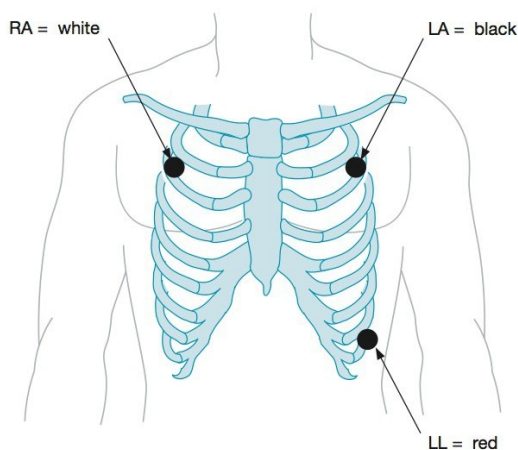
A captação do sinal pode ser feita através de eletrodos (Figura 37) conectados ao corpo, como na Figura 38. Com o sinal do ECG é possível visualizar e identificar possíveis problemas cardíacos, já que cada componente do sinal (PQRST) está associado há um movimento do coração.

Figura 37 – Eletrodos e cabo utilizados para eletrocardiograma.



Fonte: Elaboração própria.

Figura38 – Pontos de contato dos eletrodos no corpo.



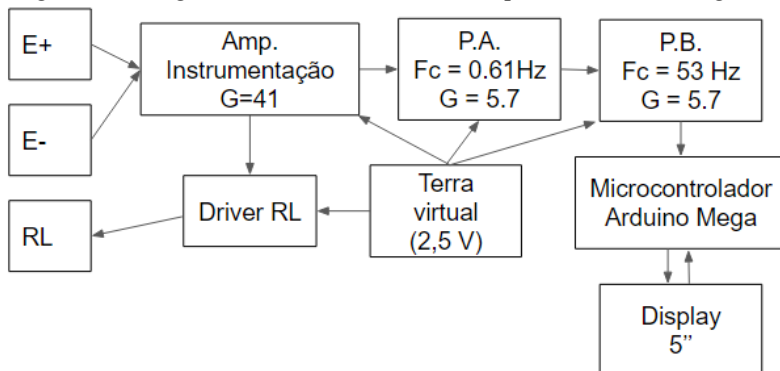
### 3 Electrode System

Fonte: Retirado de [14].

O sinal cardíaco possui amplitude típica de 0,1 mV à 5 mV e espectro de frequência de 0,2 Hz à 300 Hz. Para a sua correta instrumentação biomédica alguns circuitos são necessários, mostrados ao longo das próximas seções. Contudo, na Figura 39, é possível

visualizar bem o diagrama de blocos dos circuitos projetado para a aquisição do ECG.

Figura 39 – Diagrama de blocos do circuito completo de eletrocardiograma.



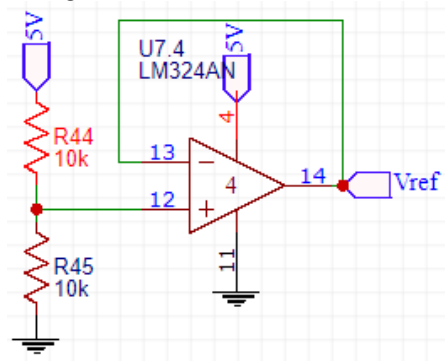
Fonte: Elaboração própria.

### 3.3.1 Circuito de terra virtual

Com o fato de que o microcontrolador utilizado trabalha com tensões de 0 a 5V, e levando em conta que o sinal cardíaco possui componentes negativas, o circuito desenvolvido precisou de um bloco de “terra virtual” (Figura 40). O bloco é formado por um divisor de tensão conectado na entrada de um buffer, disponibilizando assim em sua saída uma tensão de referência ( $V_{ref}$ ) estável de 2,5 V, conforme a Equação (3.16).

$$V_{ref} = 5 \frac{R_{45}}{R_{45} + R_{44}} = 2,5 V \quad (3.16)$$

Figura 40 – Circuito de terra virtual.



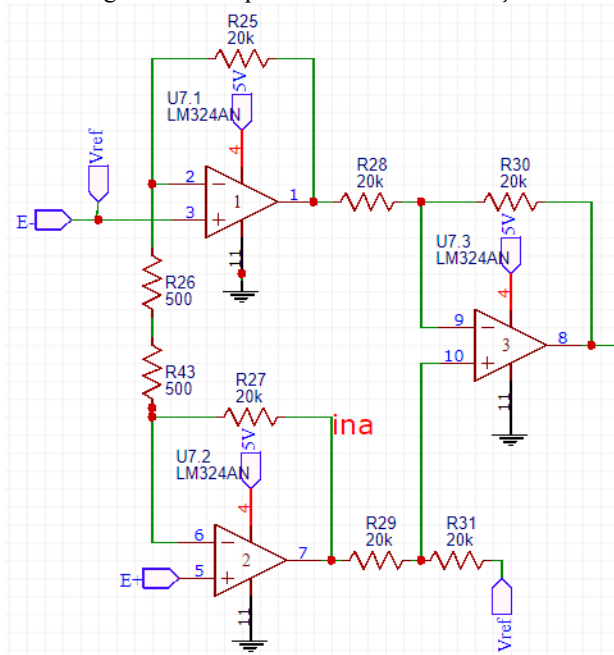
Fonte: Elaboração própria.

### 3.3.2 Amplificador de instrumentação

A captação do sinal cardíaco é feita de maneira diferencial. Inicialmente têm-se os eletrodos R (E+) e L (E-) conectados ao peito do usuário, em seguida o sinal captado precisa passar por um amplificador, porém uma versão mais simples, como o utilizado na seção 3.2.4, possui relativamente baixa imunidade à ruído, se tornando inadequado ao uso com um sinal tão complexo quanto o cardíaco. Assim optou-se por um amplificador de instrumentação (INA), desenvolvido com três AMPOPS (Figura 41) e que possui um alto CMRR (razão de rejeição de modo comum), amplificando o sinal diferencial com ganho de 41, conforme a equação abaixo.

$$\frac{V_o}{V_i} = 1 + \frac{2R}{R_g} = 1 + 2 \frac{20k}{1k} = 41 \quad (3.17)$$

Figura 41 – Amplificador de instrumentação.



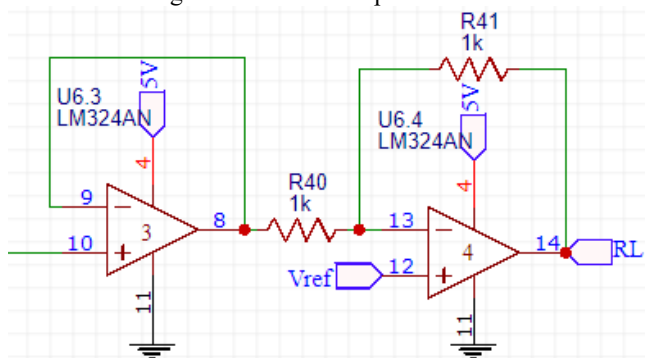
Fonte: Elaboração própria.

### 3.3.3 Circuito driver de perna direita

Para melhorar o sinal captado há um bloco de “*Driver* de perna direita” (Figura 42), que inverte o sinal comum do amplificador de instrumentação e injeta novamente no corpo do paciente, no caso não na perna direita mas sim em uma configuração igual à da Figura 38, com o intuito de que a componente comum seja somada com seu valor inverso e então eliminada.

$$VRL = -Vcom \quad (3.18)$$

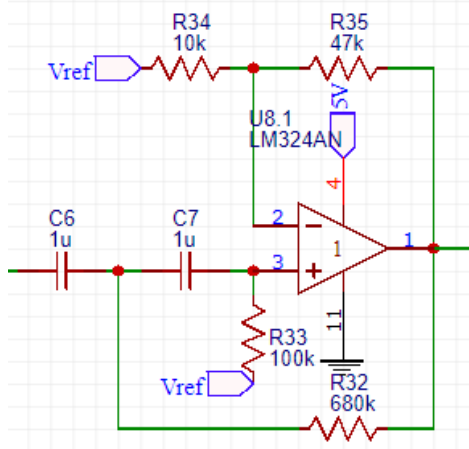
Figura 42 – Driver de perna direita.



Fonte: Elaboração própria.

### 3.3.4 Filtro passa-altas

Na saída do INA (amplificador de instrumentação) temos um filtro passa altas (Figura 43) com ganho de 5,7 e frequência de corte de 0,61 Hz, a fim de rejeitar a interferência gerada pela respiração e eliminar componentes DC indesejadas. O filtro utilizado possui uma configuração Sallen-Key, com respostas em frequência e equações iguais as da seção 3.2.5.

Figura 43 – Filtro passa-altas ativo com  $f_c = 0,61$  Hz e ganho 3.

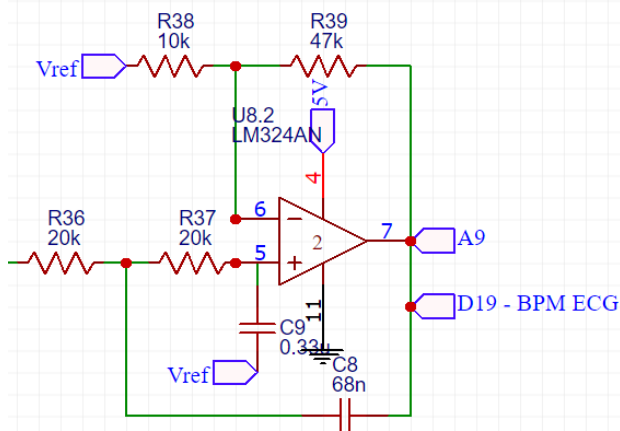
Fonte: Elaboração própria.



### 3.3.5 Filtro passa-baixas

O último estágio do tratamento do sinal é o filtro passa baixas (Figura 44) com ganho de 5,7 e frequência de corte de 53,12 Hz, para evitar a interferência do ruído da rede elétrica (60 Hz). O filtro utilizado também possui uma configuração Sallen-Key, com respostas em frequência e equações iguais as da seção 3.2.6.

Figura 44 – Filtro passa-baixas ativo com  $f_c = 53,12$  Hz e ganho 5,7.



Fonte: Elaboração própria.

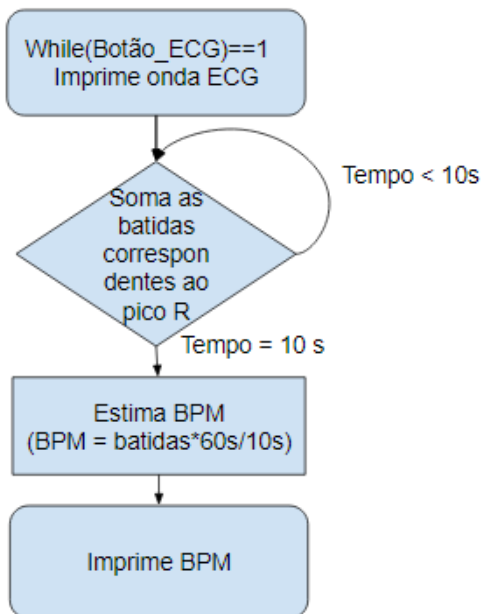


### 3.3.7 Código para aquisição do sinal cardíaco e cálculo dos BPM

O código para captação do eletrocardiograma e cálculo dos batimentos por minuto está disponível no Apêndice A, mas seu funcionamento é descrito pelo fluxograma da Figura 46.

Ao pressionar o botão relativo ao ECG, o sinal cardíaco captado começa a ser impresso no display, simultaneamente as batidas do coração são contadas durante 10 s. Isso é feito através do uso de interrupções no Arduino, que conta uma batida quando a amplitude do sinal (Pico R) passa de um determinado valor. Após o decorrer de 10 s as batidas contadas são multiplicadas por 60 e divididas por 10, para assim estimar os batimentos por minuto. Por último o resultado é impresso no *display*.

Figura 46 – Fluxograma de código para o ECG.



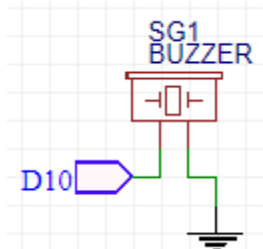
Fonte: Elaboração própria.

### 3.4 ALARME SONORO

Como forma de aviso para quando algum valor mensurado está fora da faixa considerada saudável, foi utilizado um alarme composto

por um Buzzer piezoelétrico (Figura 47) conectado à saída digital D10 do Arduino.

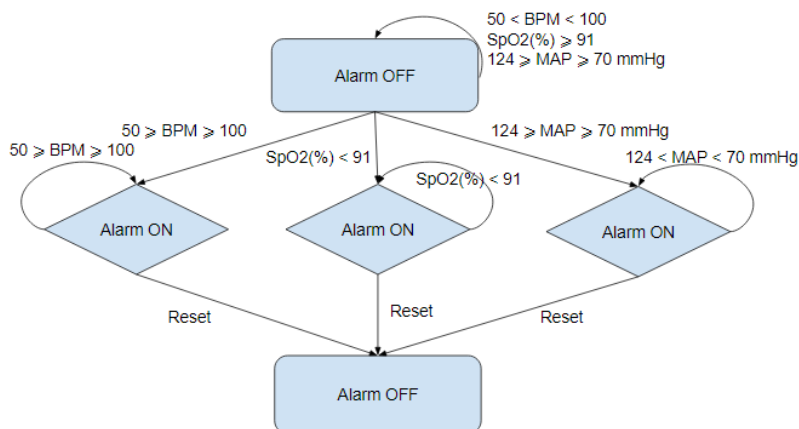
Figura 47 – Circuito de alarme com Buzzer piezoelétrico.



Fonte: Elaboração própria.

O funcionamento do alarme sonoro segue o fluxograma de código abaixo, estando inicialmente desligado e soando quando algum dos parâmetros medidos está fora das faixas da Figura 48.

Figura 48 – Fluxograma de código para o alarme.



Fonte: Elaboração própria.

### 3.5 CONECTIVIDADE COM A INTERNET

Como mencionado anteriormente o projeto tem como um de seus objetivos o uso do termo internet das coisas, integrando os dados aferidos pelos circuitos desenvolvidos com a internet. Para isso um módulo ESP8266 é utilizado, com o intuito de se conectar na rede

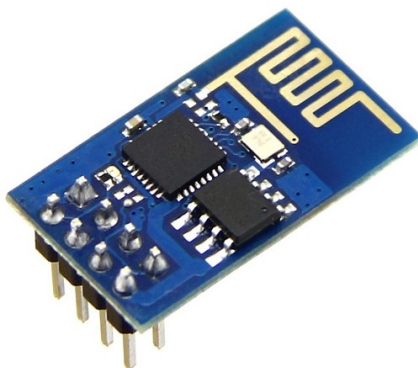
domiciliar, permitindo o envio dos dados. Os dados então são armazenados em um banco de dados online, podendo também ser visualizados tanto no site como em um aplicativo.

### 3.5.1 Módulo ESP8266

Para a conexão com a internet o módulo *wi-fi* ESP8266-01 (Figura 49) foi utilizado. Esta opção de baixo custo permitiu conectar o microcontrolador de maneira simples e eficaz à rede.

A comunicação entre o Arduino e o módulo ESP8266 é feita via porta serial através de comandos AT utilizando os pinos RX e TX. O prefixo AT vem da palavra atenção, e os comandos são uma forma básica de configurar e acionar o módulo.

Figura 49 – Módulo Wi-Fi ESP8266.



Fonte: Retirado de [15].

O ESP8266 é compacto e relativamente potente, e suas especificações podem ser vistas na tabela 5.

Tabela 5 – Especificações do módulo ESP8266.

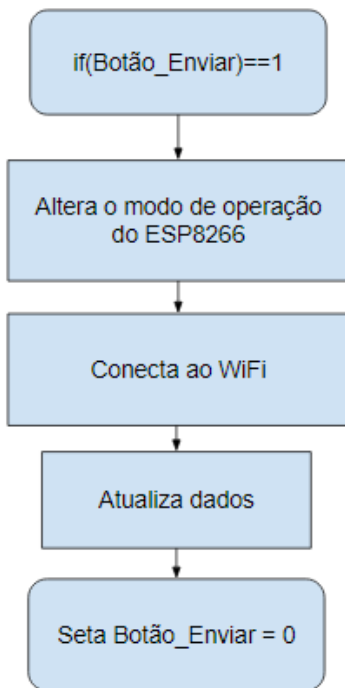
<b>Chip</b>	ESP8266
<b>Modelo</b>	ESP-01
<b>Tensão de operação</b>	3,3V
<b>Suporte à redes</b>	802.11 b/g/n
<b>Alcance</b>	90m aproximadamente
<b>Comunicação</b>	Serial (TX/RX)
<b>Suporta comunicação</b>	TCP e UDP
<b>Conectores</b>	GPIO, I2C, SPI, UART, Entrada ADC, Saída PWM e Sensor de Temperatura interno.
<b>Modo de segurança</b>	OPEN / WEP / WPA_PSK / WPA2_PSK / WPA_WPA2_PSK
<b>Dimensões</b>	25 x 14 x 1mm
<b>Peso</b>	7g

Fonte: Adaptado de [15].

No Apêndice A é possível ver o código completo para a configuração do ESP8266 e para o envio dos dados ao banco de dados. Porém o fluxograma de código abaixo foi desenvolvido para melhor compreensão do funcionamento.

Ao pressionar botão “Enviar” o módulo começa a ser configurado, alterando o seu modo de operação. Em seguida a conexão com a rede é realizada, para isso é necessário especificar no código o nome da rede e a senha. O próximo passo é o envio de dados para o banco de dados, que possui o IP da página e a chave do canal especificado no código. Após o envio dos dados, o botão é novamente setado em 0.

Figura 50 – Fluxograma de código para a conexão Wi-Fi.



Fonte: Elaboração própria.

### 3.5.2 Banco de dados

Como meio de disponibilizar e visualizar os dados coletados pelo monitor multiparamétrico, foi utilizado o site “ThingSpeak.com”. ThingSpeak é uma plataforma de serviços analíticos de IOT, que permite agregar, visualizar e analisar em tempo real dados na nuvem.

A plataforma foi escolhida por ser uma opção gratuita, que permite visualizar os dados coletados e enviados em tempo real de maneira organizada, exportar os dados diretamente para análise no MATLAB, projetar sistemas IOT sem a necessidade de configuração de servidores ou o desenvolvimento de software web. Além disso, possibilita uma fácil configuração do módulo wifi para enviar dados, através dos protocolos de IOT disponíveis na biblioteca do Arduino.

O canal <<https://thingspeak.com/channels/308474>> (Figura 51) foi criado para armazenar os dados enviados pelo monitor. Diversas opções de configuração estão disponíveis, seja sobre privacidade (canal

público ou privado) ou sobre design da página. Optou-se por deixar o canal público, a fim de demonstração, e por disponibilizar os dados em forma de gráfico, como pode ser visto na figura abaixo.

Figura 51– Página do banco de dados utilizado.

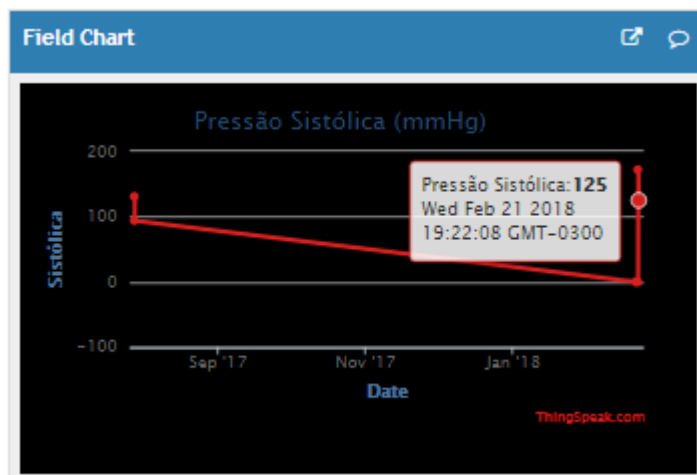


Fonte: Elaboração própria.

Em cada gráfico constam os valores recebidos no eixo y e a data no eixo x. Ao clicar em cada ponto do gráfico (Figura 52) é possível visualizar o valor obtido e a data precisa que o dado foi recebido. No primeiro gráfico constam os valores enviados dos batimentos por minuto medidos com o ECG, no segundo constam os valores da porcentagem saturação de oxigênio (SpO2), no terceiro consta a pressão sistólica em mmHg e no quarto a pressão diastólica também em mmHg.



Figura 52 – Gráfico da pressão sistólica disponível na página.



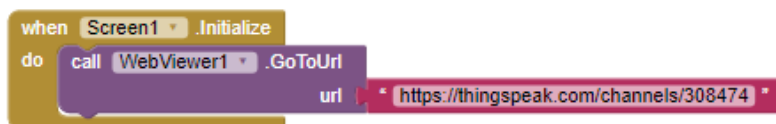
Fonte: Elaboração própria.

### 3.5.3 Aplicativo

Embora a página possa ser acessada através de qualquer navegador de internet padrão, um aplicativo foi desenvolvido para Android (Figura 54), pensando em uma maior comodidade para o usuário acessar os dados.

O desenvolvimento foi feito na plataforma MIT App Inventor, que permite o projeto de aplicativos funcionais para smartphones e tablets Android, através de um ambiente de programação visual, onde a codificação é feita por blocos, como o da Figura 53.

Figura 53 – Programação por blocos para o aplicativo desenvolvido.

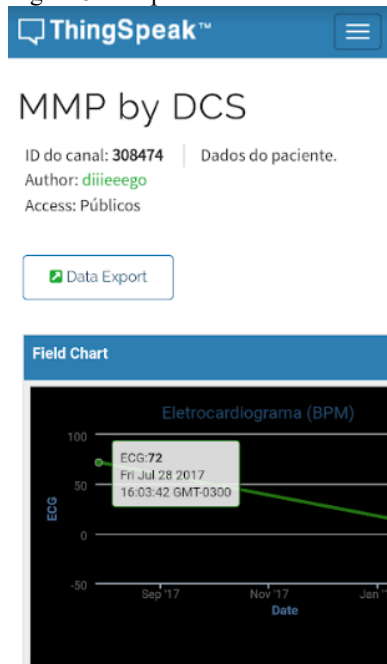


Fonte: Elaboração própria.

O aplicativo funciona de forma simples, atuando como um facilitador para o usuário. Ao abri-lo, a página contendo os dados online

do paciente será automaticamente carregada, permitindo assim um acesso mais rápido aos valores coletados.

Figura 54 – Aplicativo desenvolvido.



Fonte: Elaboração própria.

### 3.6 DISPLAY

Como forma primária de exposição dos gráficos e resultados das medições optou-se por utilizar um display. Tendo em mente que o objetivo era desenvolver um equipamento para uso domiciliar, que fosse compacto e portátil, um display tátil de cinco polegadas (Figura 55) foi escolhido, por possuir um tamanho relativamente pequeno mas com uma boa resolução para ver os resultados, e por ainda contar com as facilidades do *touchscreen*.

Figura 55 – Display táctil de 5 polegadas utilizado.



Fonte: Elaboração própria.

O modelo escolhido foi o módulo Display SSD1963, com resolução de 800X480, painel *touch*, e entrada para cartão SD. Optou-se por ele devido as suas especificações (Tabela 6) e custo benefício.

Tabela 6 – Especificações do Display.

<b>Tamanho da tela</b>	5.0 (Diagonal)
<b>Formato do display</b>	800 GRB x 480
<b>Área ativa</b>	108°
<b>Pixel Pitch</b>	0.135 (H) X 0.135 (V)
<b>Tratamento da superfície</b>	Anti-glare
<b>Configuração de Pixel</b>	RGB-Stripe
<b>Dimensão do traçado</b>	118.5 (H) X 77.55 (V) X 3.4 (T)
<b>Dimensões:</b>	13.5 cm X 8.5cm / 5.31" x 3.34" (LxW) (Aproximadamente)

Fonte: Adaptado de [16].

Como o display utilizado trabalha com tensão de 3,3V e as entradas e saídas do Arduíno operam com 5 V, é necessário um módulo para converter as tensões. Para isso foi utilizador o módulo da Figura 56.

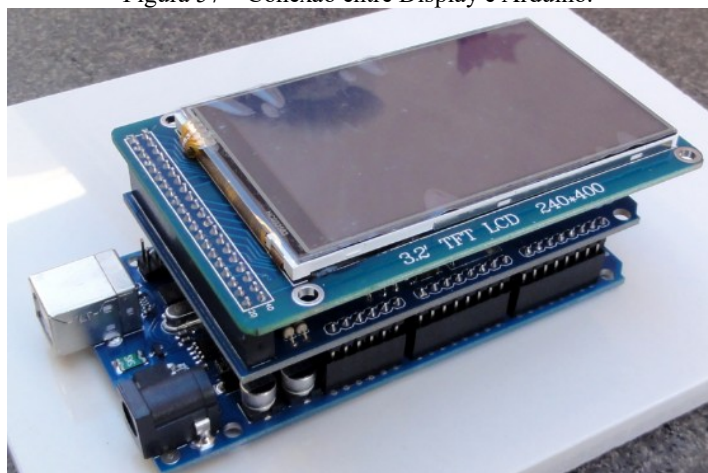
Figura 56 – *Shield* regulador de tensão para o Display.



Fonte: Retirado de [17].

O módulo (*shield*) conversor de tensão atua como uma interface entre o display e o Arduino Mega, e a conexão dos três itens é similar ao exemplo da Figura 57.

Figura 57 – Conexão entre Display e Arduino.

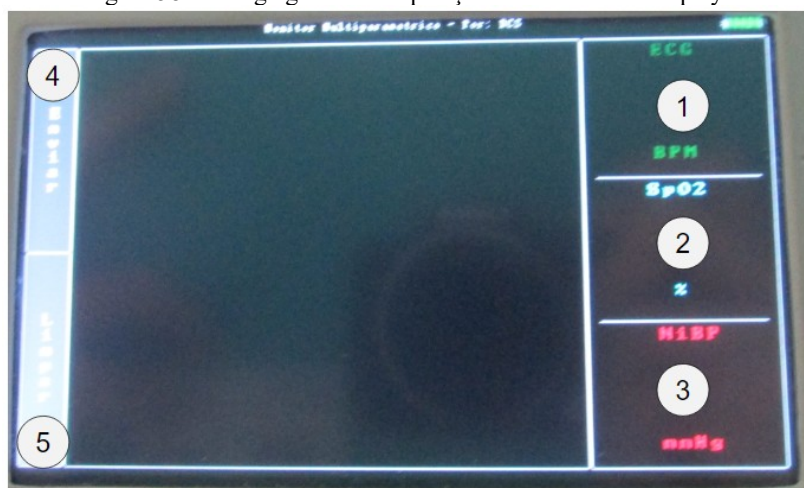


Fonte: Retirado de [18].

O display utilizado possui duas bibliotecas disponíveis para o auxílio na programação com o Arduino. Uma para a configuração da imagem na tela e uma para o *touch screen*. Todos os códigos podem ser vistos no Apêndice A, porém o funcionamento do display é descrito pelas Figuras 58 e 59 e pela Tabela 7.

O display foi projetado para possuir 5 botões com as funções descritas na tabela 7.

Figura 58 – Design gráfico e disposições dos botões no display.



Fonte: Elaboração própria.

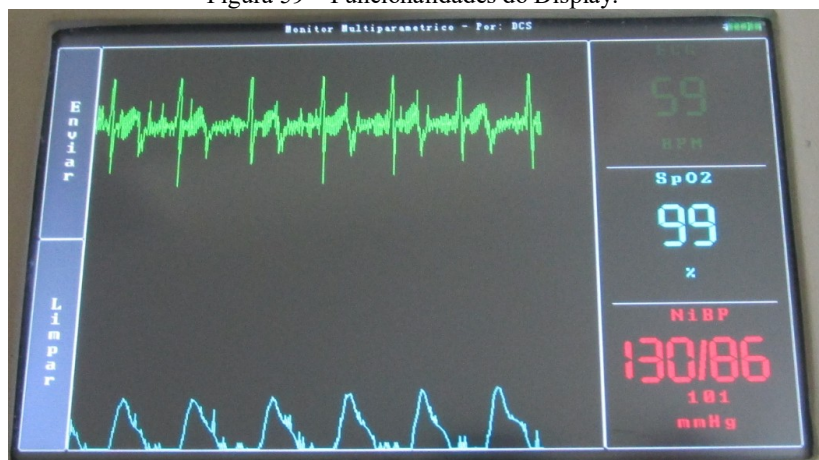
Tabela 7 – Funcionalidade dos botões *touchscreen* do *display*.

Numero	Botão	Funcionalidade
1	ECG	Imprime o ECG e os batimentos por minuto.
2	SpO2	Imprime o gráfico do PPG e a porcentagem de saturação de oxigênio
3	NiBP	Inicia o processo de medição da pressão arterial e imprime as pressões sistólica e diastólica.
4	Enviar	Envia os resultados para a nuvem.
5	Limpar	Limpa os gráficos e medidas.

Fonte: Elaboração própria.

Quando os botões 1 e 2 são pressionados, além da exposição das medidas de BPM e SpO2 (%), os gráficos de ECG e PPG são impressos em tempo real, quase que simultaneamente (Figura 59). Quando o botão 3 é selecionado as demais funções param devido às limitações de processamento do microcontrolador. O processo de medição da pressão não invasiva começa. Quando o resultado está pronto ele é impresso no campo correspondente e as demais medidas são retomadas. A mesma interrupção ocorre quando é pressionado o botão 4, que envia os valores de pressão, BPM e saturação para o banco de dados.

Figura 59 – Funcionalidades do Display.



Fonte: Elaboração própria.

### 3.7 ALIMENTAÇÃO DO SISTEMA

O monitor multiparamétrico foi desenvolvido com o intuito de ser algo compacto e portátil, pensando nisso é necessário o uso de uma bateria, bem como circuitos que regulem a tensão oferecida, tendo em vista que algumas partes do projeto operam com tensões diferentes.

#### 3.7.1 Bateria

Como fonte de alimentação para os componentes foi utilizada uma bateria SAMSUNG de Li-ion (Figura 60), que possui tensão de 3,8V e fornece até 1900 mAh (7.22 Wh), sendo escolhida por ser uma boa opção de bateria recarregável, possuindo uma boa capacidade e bom custo-benefício.

Figura 60 – Bateria de Li-ion de 3,8 V 1900 mAh.



Fonte: Elaboração própria.

#### 3.7.2 Circuito de carga

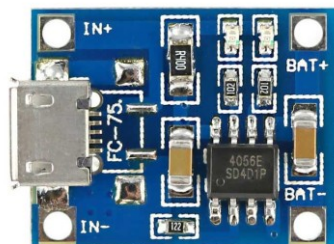
Para recarregar a bateria é utilizado o módulo carregador TP4056 (Figura 61), que possui entrada micro USB, fornece 4,2 V e 1 A à bateria, e possui LEDs vermelho indicando carregamento e verde para carregado. As especificações do módulo podem ser vistas na tabela abaixo.

Tabela 8 – Especificações do módulo carregador de bateria TP4046.

<b>Chip</b>	TP4056
<b>Compatibilidade</b>	Baterias de Li-ion
<b>Tensão de entrada</b>	4,5 V ~ 5,5V
<b>Tensão totalmente carregada</b>	4,2 V
<b>Corrente de carregamento</b>	1 A ajustável
<b>Temperatura de trabalho</b>	-10°C a 85°C
<b>Método de carregamento</b>	Carga linear 1%
<b>Precisão de carga</b>	1,5 %
<b>Interface</b>	Micro USB
<b>Dimensões</b>	23,5 x 17 x 4 mm
<b>Peso</b>	1,5 g

Fonte: Adaptado de [19].

Figura 61 – Módulo carregador de bateria TP4046.



Fonte: Retirado de [20].

### 3.7.3 Regulador de tensão *Boost Step Up*

Tendo em vista que a bateria possui tensão de 3,8 V, e que o Arduino Mega 2560 necessita de no mínimo 7 V em sua entrada (antes do regulador de tensão interno da placa), é necessário a implementação de um circuito que aumente a tensão da bateria fornecendo tensão e corrente estáveis. Para isso é utilizado o módulo regulador de tensão ajustável MT3608 *Step UP* (Figura 62), ajustado para fornecer uma tensão fixa na saída de 7 V. A Tabela 9 exhibe as especificações técnicas do módulo.



Tabela 9 – Especificações do módulo regulador de tensão Step-Up.

<b>Tensão de entrada</b>	2 V ~ 24 V
<b>Tensão de saída ajustável</b>	2,5 V ~ 28 V
<b>Regulação de carga</b>	+/- 0,5 %
<b>Eficiência</b>	91 %
<b>Diferença mínima entre in e out</b>	0,5 V
<b>Frequência de comutação</b>	150 KHz
<b>Corrente máxima</b>	2 A
<b>Dimensões (CxLxA)</b>	37x17x6mm;
<b>Peso</b>	4 g

Fonte: Adaptado de [21].

Figura 62 – Regulador de tensão ajustável MT3608 *Step Up*.



Fonte: Retirado de [22].

### 3.7.4 Reguladores de tensão de 3 V e 5 V

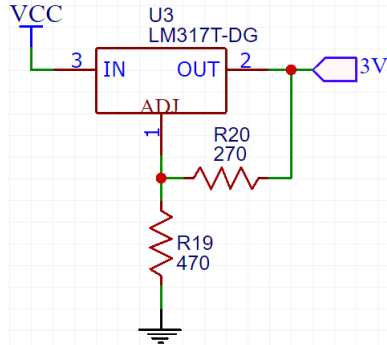
Embora a bateria seja de 3,8 V e o Arduino seja alimentado por uma tensão de 7 V, o circuito do monitor possui componentes que operam com 5 V (CIs e circuitos de instrumentação) e componentes que necessitam de 3 V (micro bomba de ar e válvula solenóide). Portanto os dois reguladores de tensão das Figuras 63 e 64 precisaram ser projetados.

Para fornecer 3 V, o regulador LM317 foi utilizado com tensão ajustável através de uma configuração básica com dois resistores, cuja tensão de saída é dada pela equação (3.19) onde  $V_{ref}$  é 1,25 V e  $I_{adj}$  é

tipicamente 50 uA, não contribuindo de maneira significativa ao valor final.

$$V_o = V_{ref} \left( 1 + \frac{R_2}{R_1} \right) + (I_{adj} * R_2) \quad (3.19)$$

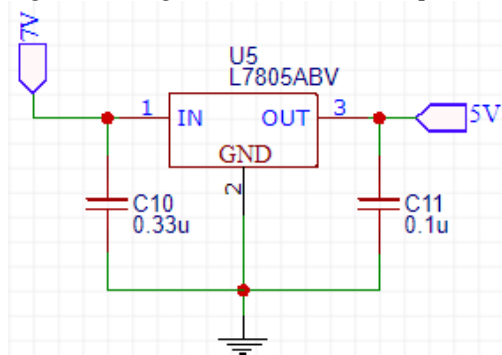
Figura 63 – Regulador de tensão de 7 V para 3V.



Fonte: Elaboração própria.

Já para fornecer 5 V para os circuitos de instrumentação, o regulador LM7805 foi utilizado, com tensão de saída de 5 V obtida diretamente para tensões de entrada de 7~20 V. Os capacitores não são obrigatórios, mas o capacitor de 0,33 uF na entrada é utilizado para garantir uma melhor imunidade ao ruído e melhorar a estabilidade, e o capacitor de saída de 0,1 uF é utilizado para melhorar a resposta ao transiente, conforme [23].

Figura 64 – Regulador de tensão de 7 V para 5V.



Fonte: Elaboração própria.

### 3.8 DESENHO DA PLACA DE CIRCUITO IMPRESSO

Com todo circuito do monitor multiparamétrico devidamente projetado, simulado e testado, a próxima etapa foi a confecção da placa de circuito impresso (PCI). O *layout* do circuito foi desenvolvido através da plataforma online EasyEDA. Abaixo podem ser vistas imagens do *layout* final da placa (Figura 65) e da disposição dos componentes (Figura 66).

Figura 65 – Layout da placa de circuito impresso desenvolvida.

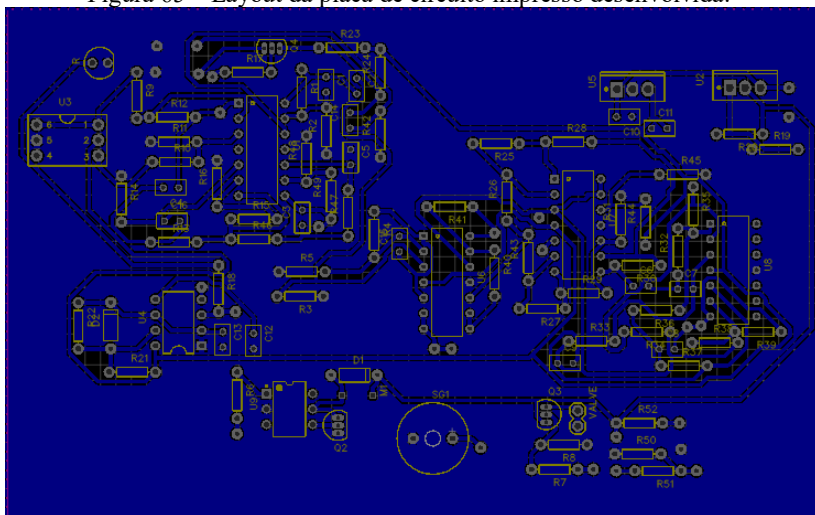
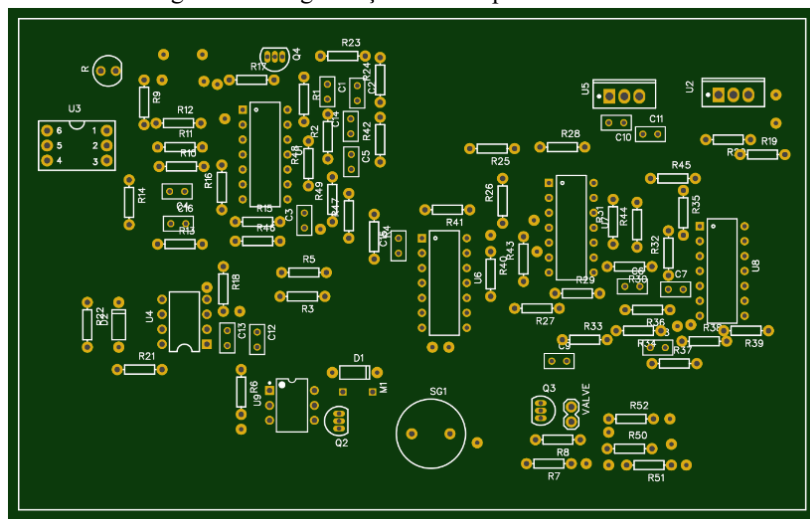


Figura 66 – Organização dos componentes na PCI.

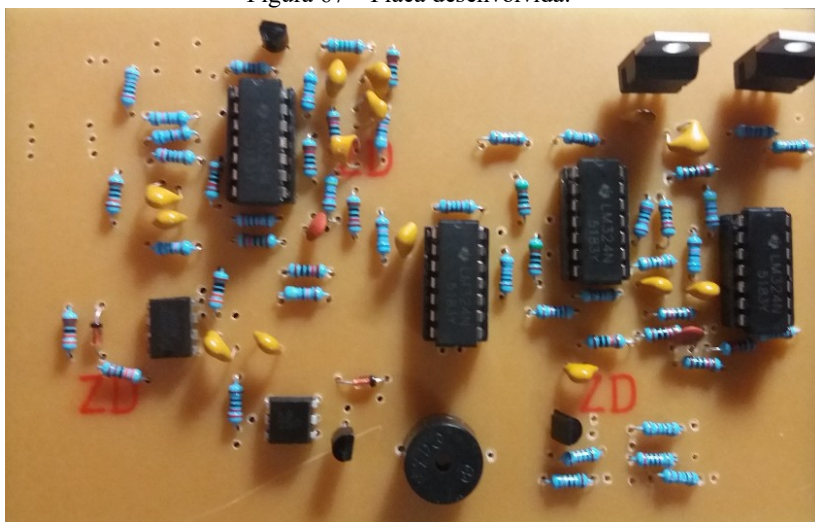


Fonte: Elaboração própria.

A confecção da placa foi feita manualmente através do método térmico, onde inicialmente o layout espelhado é impresso com impressora a laser em papel com gramatura grande. Depois a folha é colocada sobre a superfície de cobre da placa, então a placa é aquecida para que a tinta grude bem. O último passo é colocar a placa em uma solução de água com percloro de ferro anidro. Quando em contato com a água o percloro sofre hidrólise, liberando calor e corroendo o cobre, deixando assim apenas as trilhas impressas no papel que foram grudadas na placa.

Com a PCI devidamente confeccionada, o próximo passo foi a inserção dos componentes e a soldagem deles na placa. Na figura 67 é possível ver a PCI montada.

Figura 67 – Placa desenvolvida.

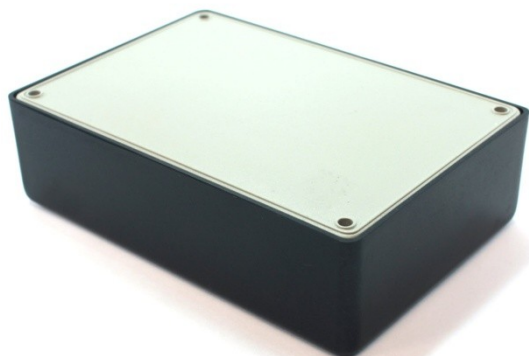


Fonte: Elaboração própria.

### 3.9 DESIGN FINAL DO EQUIPAMENTO

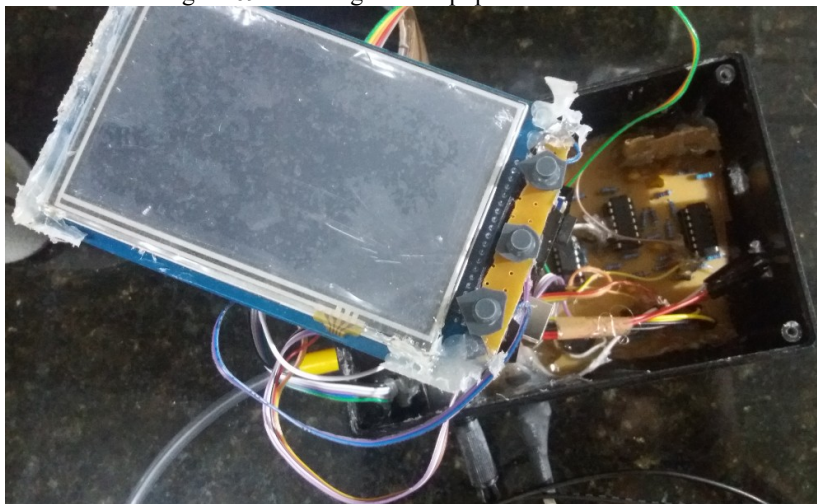
Para o design final do equipamento buscou-se utilizar o tamanho mais compacto possível. Para isso optou-se por utilizar uma caixa plástica de prototipagem Patola (Figura 68) com dimensões de 55 x 97 x 148 mm, tamanho mínimo para acomodar todos os componentes, como pode ser visto na figura 69.

Figura 68 – Caixa Patola / Case para Montagem 55 x 97 x 148 mm - PB-114.



Fonte: Retirado de [24].

Figura 69 – Montagem do equipamento na caixa.

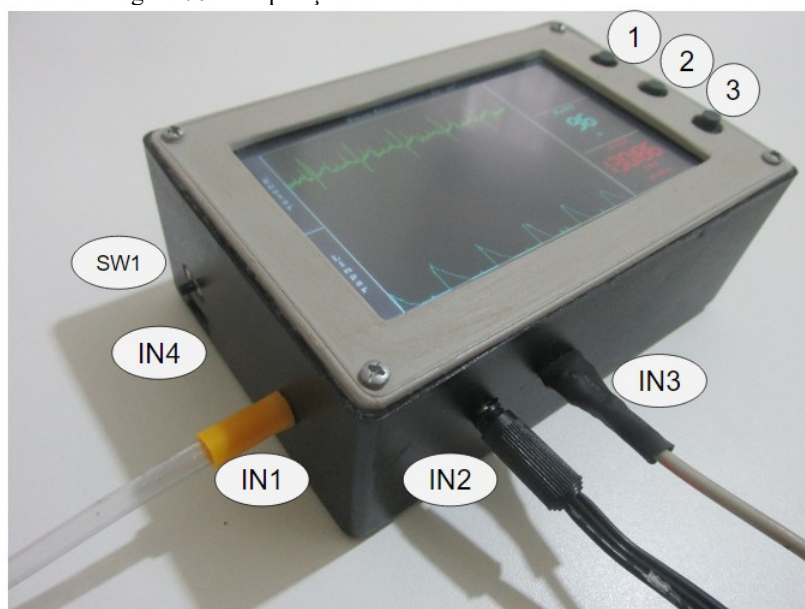


Fonte: Elaboração própria.

O equipamento foi projetado para ter um design simples e de fácil utilização. Para isso além das funções de *touch* do *display*, três botões (Figura 70) foram implementados a direita do equipamento, com as

mesmas funcionalidades dos botões 1, 2 e 3. Além de ser escolhida uma localização adequada para as entradas da mangueira do manguito (IN1), cabo dos eletrodos do ECG (IN2), sensor de oximetria de pulso (IN3), entrada micro USB para carregar a bateria (IN4), entrada alternativa para alimentação (IN5 – Figura 71), entrada para programação do Arduino (IN6 – Figura 71) e *switch* (SW1) para ligar e desligar o monitor multiparamétrico.

Figura 70 – Disposição dos botões e conexões no monitor.



Fonte: Elaboração própria.

Figura 71– Disposição das conexões no monitor.



Fonte: Elaboração própria.

Nas Figuras 72 e 73 é possível observar melhor o design final do monitor multiparamétrico portátil com conexão *Wi-Fi* projetado.

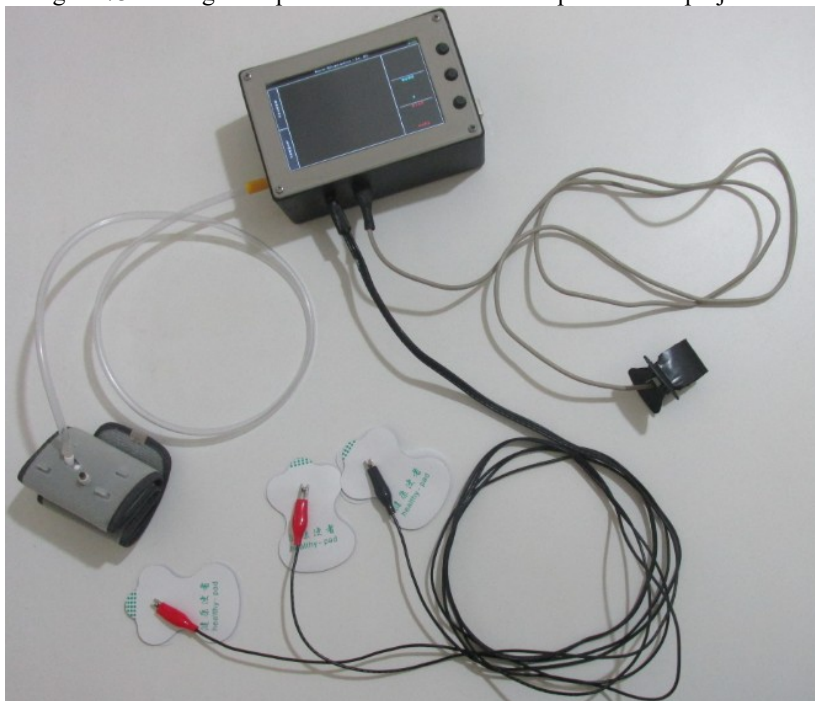
Figura 72 – Design frontal final do monitor projetado.



Fonte: Elaboração própria.



Figura 73 – Design completo final do monitor multiparamétrico projetado.



Fonte: Elaboração própria.

## 4 AVALIAÇÃO DO PROTÓTIPO

Diversos parâmetros são importantes na análise dos resultados e do desempenho do equipamento projetado. Podendo ser divididos em dois tipos: avaliação de custos e desempenho, onde se encontram consumo, duração da bateria, custo total do equipamento, etc. E validação, onde se enquadram os gráficos dos sinais fisiológicos, proximidade das medidas obtidas com os valores reais e etc.

### 4.1 AVALIAÇÃO DE CUSTOS E DESEMPENHO

Na avaliação de custos e desempenho do equipamento buscou-se averiguar os limites do monitor desenvolvido.

Com a bateria completamente carregada testou-se a duração média, máxima e mínima, bem como o consumo do MMP projetado, chegando assim à Tabela [10]. Onde os valores de duração mínima e

consumo máximo foram obtidos com a micro bomba de ar ligada. Duração máxima e mínima foram obtidos com o equipamento ligado, mas sem estar realizando nenhuma medida. E a duração média e consumo médio foram aferidos com o equipamento captando o ECG e SpO2.

Tabela 10 – Consumo e duração da bateria.

<b>Parâmetro</b>	<b>Resultado</b>
Duração mínima	00:53 hr
Duração média	01:07 hr
Duração máxima	01:15 hr
Consumo mínimo	1520 mA , 5,8 W
Consumo médio	1700 mA, 6,5 W
Consumo máximo	2150 mA, 8,2 W

Fonte: Elaboração própria.

Outro ponto importante analisado foi o tempo para obtenção das pressões diastólica e sistólica, da porcentagem de saturação de oxigênio e dos batimentos por minuto. O tempo medido para a pressão foi tomado como um valor médio, tendo em vista que a pressão na coifa pode variar de acordo com o diâmetro do pulso do usuário, portanto levando mais ou menos tempo para inflar ou esvaziar. Já para a SpO2 os dados estão disponíveis de segundo em segundo, porém são atualizados no display apenas quando os gráficos atingem o limite esquerdo da tela, devido a limitações de processamento do micro controlador, levando assim o mesmo tempo que o gráfico leva para ser completo para disponibilizar o resultado. Por último os batimentos por minuto são calculados como uma estimativa, assim os dados são coletados apenas durante 10 segundos, porém são atualizados no display da mesma maneira que a SpO2, apenas quando os gráficos atingem o limite esquerdo. Esses resultados podem ser vistos na tabela [11].

Tabela 11 – Tempo médio para obtenção das medidas.

<b>Parâmetro</b>	<b>Resultado</b>
NiBp	90 s
Bpm	12 s
SpO2	6 s

Fonte: Elaboração própria.

O tempo que o dispositivo leva para enviar um parâmetro e atualizar o banco de dados também foi analisado, tendo em vista que é uma característica importante para o equipamento. O resultado pode variar se uma conexão com a internet não for estabelecida corretamente, ou caso a conexão caia. Contudo, normalmente não ocorrem quedas, sendo assim o tempo médio para envio e atualização do banco de dados é o da tabela [12].

Tabela 12 – Tempo médio para o envio via Wi-Fi dos dados.

<b>Parâmetro</b>	<b>Resultado</b>
Tempo médio para envio e atualização do banco de dados	60 s

Fonte: Fonte: Elaboração própria.

Para a análise do custo do equipamento não foram levados em conta o salário por hora trabalhada do engenheiro, o custo de componentes se comprados em larga escala e nem o preço final para venda no mercado. Na tabela [13] é possível observar os valores dos componentes individuais adquiridos para o desenvolvimento do projeto, bem como o custo total.

Tabela 13 – Custo individual dos componentes e total do produto.

<b>Componente</b>	<b>Preço</b>
Arduino Mega 2560	R\$ 45,50
<i>Display 5" Touchscreen</i>	R\$ 122,32
<i>Shield</i> regulador para o <i>Display</i>	R\$ 21,70
Módulo carregador de bateria	R\$ 5,80
Módulo regulador <i>boost SteepUp</i>	R\$ 9,90
Módulo <i>Wi-Fi</i> ESP8266	R\$ 8,92
Resistores, capacitores, transistores, LED's, reguladores, soquetes e CI's.	R\$ 41,68
Placa fenolite	R\$ 4,90
Caixa patola	R\$ 22,90
Micro bomba de ar	R\$ 19,69
Válvula solenóide	R\$ 4,15
Sensor de pressão	R\$ 27,90
<b>Total:</b>	R\$ 335,36

Fonte: Elaboração própria.

Outras características importantes são o peso e as dimensões do equipamento, tendo em vista que o intuito é que ele seja portátil. Esses parâmetros podem ser vistos na tabela 14.

Tabela 14 – Dimensões finais e peso.

<b>Parâmetro</b>	<b>Resultado</b>
Dimensões	55 x 97 x 148 mm
Peso	500 g

Fonte: Elaboração própria.

## 4.2 VALIDAÇÃO

Buscando obter um bom parâmetro para comparar o monitor desenvolvido com um equipamento comercial, e visando a realização de medidas sem a necessidade de testes em seres humanos ou animais, uma série de dados foi coletada em manequins eletrônicos, disponíveis no Laboratório de Práticas Simuladas (LPS), situado no Centro de Ciências da Saúde da Universidade Federal de Santa Catarina (CCS – UFSC).

O LPS possui um ambiente que simula três leitos hospitalares, com equipamentos médicos típicos, e com manequins eletrônicos para o uso em simulações clínicas de alta fidelidade, conhecidos como HPS “*Human Patient Simulator*”, sendo anatomicamente precisos e simulando respostas fisiológicas humanas.

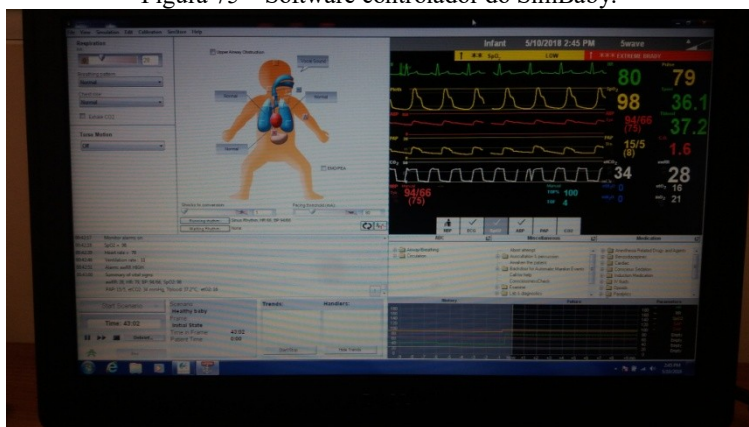
Para o presente estudo o HPS SimBaby (Figura 74), da empresa Laerdal, foi utilizado. Ele é um simulador avançado de paciente recém-nascido para treinamento em equipe, possuindo anatomia real e funcionalidade clínica. Seu controle é feito através de um computador conectado ao sistema, o qual possui programas de criação e edição de cenários. A programação é intuitiva e feita por blocos e utiliza um software exclusivo para rodar o cenário (Figura 75), onde é possível realizar as simulações pré-programadas ou até mesmo alterar instantaneamente os valores desejados.

Figura 74 – SimBaby.



Fonte: Retirado de [25].

Figura 75 – Software controlador do SimBaby.



Fonte: Elaboração própria.

O SimBaby possui uma série de parâmetros que podem ser configurados, desde a simulação de estado normal do paciente, até uma convulsão, arritmia cardíaca, etc. Porém para o efetivo teste do equipamento projetado, conectou-se os cabos do ECG conforme a Figura 76, o sensor de oximetria de pulso no dedo e a braçadeira para a pressão no pulso do manequim. Simultaneamente o monitor multiparamétrico do leito foi conectado ao “paciente” e então uma simulação de estado “padrão” foi rodada.

Figura 76 – SimBaby.



Fonte:Elaboração própria.

Ao longo da simulação foram configurados no software 10 valores distintos para cada parâmetro de interesse. Para a saturação de oxigênio e para o eletrocardiograma foram realizadas 3 avaliações para cada valor configurado no simulador, enquanto que para a pressão arterial foi realizada uma avaliação para cada valor, por questões de tempo. Os resultados encontram-se nas Tabelas 15, 16 e 17.

Tabela 15 – Valores obtidos para BPM.

Parâmetro	Valor configurado no Software de controle	Resultado no MMP do LPS	Resultado no MMP projetado
BPM	40 bpm	39 – 40 – 39 bpm	37 – 41 – 41 bpm
	45 bpm	44 - 45 - 45 bpm	42 – 45 – 45 bpm
	50 bpm	49 – 49 – 50 bpm	48 – 49 – 49 bpm
	55 bpm	55 - 54 -54 bpm	57 – 52 – 52 bpm
	60 bpm	58 – 58 – 60 bpm	58 – 58 - 61 bpm
	65 bpm	64 – 64 – 63 bpm	63 – 63 – 60 bpm
	70 bpm	67 – 69 – 70 bpm	71 – 72 – 72 bpm
	75 bpm	73 – 74 – 74 bpm	72 – 74 – 74 bpm
	80 bpm	79 – 79 -81 bpm	77 – 79 – 79 bpm
	85 bpm	84 – 84 – 85 bpm	85 – 85 – 89 bpm

Fonte: Elaboração própria.

Tabela 16 – Valores obtidos para a pressão arterial.

<b>Parâmetro</b>	<b>Valor configurado no Software de controle</b>	<b>Resultado no MMP do LPS</b>	<b>Resultado no MMP projetado</b>
NiBp	70/40 mmHg	71/41 mmHg	65/43 mmHg
	80/50 mmHg	80/51 mmHg	78/52 mmHg
	90/55 mmHg	86/54 mmHg	87/58 mmHg
	100/60 mmHg	96/58 mmHg	96/64 mmHg
	110/70 mmHg	106/69 mmHg	110/73 mmHg
	120/80 mmHg	121/82 mmHg	125/83 mmHg
	130/85 mmHg	124/82 mmHg	131/87 mmHg
	140/90 mmHg	143/92 mmHg	141/94 mmHg
	150/100 mmHg	149/100 mmHg	145/97 mmHg
	160/110 mmHg	160/115 mmHg	157/104 mmHg

Fonte: Elaboração própria.

Tabela 17 – Valores obtidos para a saturação de oxigênio.

<b>Parâmetro</b>	<b>Valor configurado no Software de controle</b>	<b>Resultado no MMP do LPS</b>	<b>Resultado no MMP projetado</b>
SpO2	90%	89 - 89 -90 %	91 – 93 – 93 %
	91%	91 – 92 -92 %	91 – 91 - 93 %
	92%	91 – 93 – 93 %	91 – 91 – 93 %
	93%	93 – 93 - 92 %	94 – 93 – 95 %
	94%	93 – 95 – 95 %	93 – 93 – 94 %
	95%	94 – 95 – 95 %	95 – 93 – 96 %
	96%	95 – 96 – 97 %	95 – 95 – 96 %
	97%	96 – 96 – 98 %	96 – 97 – 99 %
	98%	98 – 98 - 98 %	98 – 98 – 97 %
	99%	98 - 99 - 98%	98 – 99 – 99 %

Fonte: Elaboração própria.





## 5 CONSIDERAÇÕES FINAIS

Com todos esses conhecimentos e a utilização de circuitos adequados foi possível implementar o projeto, coletando os dados do parâmetro selecionado pelo paciente e os disponibilizando via *wi-fi* em um banco de dados, que futuramente poderá ser utilizado em avaliações médicas ou por empresas especializadas.

Analizando os resultados obtidos nos testes percebe-se que o consumo do circuito ficou um pouco elevado, fazendo com que a bateria dure em média 01:07 hrs. Isso pode ser melhorado em projetos futuros, através da implementação do controle de brilho do display e de um algoritmo de controle para o módulo *Wi-Fi* e para os circuitos de instrumentação, alimentando-os apenas quando o botão correspondente for ativado.

Em relação à usabilidade do equipamento, acredita-se ter atingido o objetivo, chegando na versão final com dimensão e peso bastante reduzidos (55 x 97 x 148 mm e 500 g), o tornando portátil. Além de ter atingido uma interface de usuário simples e amigável, tanto com o display e botões, quando com o banco de dados e o aplicativo, obtendo-se ainda tempos aceitáveis para a realização das medições.

Um fato importante a ser mencionado é a segurança dos dados do paciente, que embora durante a documentação não tenha sido abordado nenhuma criptografia ou padrão de dados para a representação e transferência de dados clínicos, como o *Health Level 7* (HL7), por exemplo, há um projeto de mestrado de uma aluna no IEB que busca implementar tais padrões, utilizando o monitor multiparamétrico aqui projetado como equipamento base.

Com a realização da série de testes feitos no manequim robótico foi possível averiguar que o MMP projetado realiza corretamente as medições dos parâmetros fisiológicos desejados. Embora possua uma variação média maior nos valores medidos (1,6 BPM; 2,95 mmHg; 0,83 SpO2%) se comparado com o monitor do LPS (0,93 BPM; 2,1 mmHg; 0,63 SpO2%), ainda atinge um resultado bastante satisfatório.

Pode-se dizer que o objetivo do projeto foi atingindo, realizando a implementação do Monitor Multiparamétrico com *Wi-Fi* para uso portátil e domiciliar. Obtendo-se ainda um design final bastante amigável e portátil, com um custo de componentes bem reduzido, mesmo que comprados em lugares diferentes e em pequenas quantidades (R\$ 335,36). Vale destacar que equipamentos com finalidades e especificações técnicas semelhantes, porém sem conexão

com internet, possuem valores comerciais no Brasil em torno de R\$ 6000,00. Embora tenha-se o conhecimento que tais equipamentos são testados e aprovados por órgãos reguladores e possuam maior qualidade de componentes e dos dados coletados. Ainda assim acredita-se que o resultado alcançado foi bastante satisfatório, se tratando de um equipamento desenvolvido em um curto período de tempo, com poucos recursos e por uma única pessoa.

Contudo, o projeto desenvolvido proporcionou a utilização de conceitos aprendidos ao longo de todo o curso de graduação em Engenharia Eletrônica, principalmente aqueles relacionados às matérias na área de Engenharia Biomédica. Acredita-se que o equipamento atingiu os resultados esperados, sendo que a experiência adquirida durante os meses de desenvolvimento foi de fundamental importância para a minha formação profissional.





## REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Imagem do monitor multiparamétrico. Disponível em:  
<http://www.bemed.com.br/imagens/produtos/imagens-lupa/monitor-12-monitor-multiparametrico.jpg> - Acessado em 11/08/2017.
- [2] Ficha técnica do Arduino Mega. Disponível em:  
<http://www.baudaeletronica.com.br/arduino-mega.html> - Acessado em 17/05/2018.
- [3] Ciclo PDCA. Disponível em:  
<https://socialconcursos.com.br/blog/ciclo-pdca-definicao> - Acessado em 17/05/2018.
- [4] Artigo sobre saturação de oxigênio. Disponível em:  
<http://www.ti.com/lit/an/slaa655/slaa655.pdf> - Acessado em 17/05/2018.
- [5] Ferramenta para projeto de filtros passivos. Disponível em:  
<http://sim.okawa-denshi.jp/en/CRtool.php> - Acessado em 17/05/2018.
- [6] Ferramenta para projeto de filtros ativos. Disponível em:  
<http://sim.okawa-denshi.jp/en/optool.php> - Acessado em 17/05/2018.
- [7] Curva característica para a pressão arterial. Disponível em:  
<http://tensoval.com/oscillometric-measurement-devices.php> - Acessado em 17/05/2018.
- [8] Microbomba de ar. Disponível em:  
<https://he.aliexpress.com/item/Micro-DC-air-pump-for-DIY-toys-as-an-increasing-oxygen-pump-in-fish-tank-free/600909126.html> - Acessado em 17/05/2018.
- [9] Válvula solenóide. Disponível em:  
[https://ae01.alicdn.com/kf/HTB1f\\_VZLVXXXXX7XFXXq6xXFXXXC/DC-3V-Apply-for-Sphygmomanometer-Electronic-font-b-Blood-b-font-Pressure-Solenoid-Valve-Release-Medical.jpg](https://ae01.alicdn.com/kf/HTB1f_VZLVXXXXX7XFXXq6xXFXXXC/DC-3V-Apply-for-Sphygmomanometer-Electronic-font-b-Blood-b-font-Pressure-Solenoid-Valve-Release-Medical.jpg) - Acessado em 17/05/2018.
- [10] Sensor de pressão. Disponível em:  
[https://softroboticstoolkit.com/files/sorotoolkit/files/mps20n0040d-s\\_datasheet.pdf](https://softroboticstoolkit.com/files/sorotoolkit/files/mps20n0040d-s_datasheet.pdf) - Acessado em 17/05/2018.

- [11] Características do sensor de pressão. Disponível em: <https://www.usinainfo.com.br/sensor-de-pressao-arduino/sensor-de-pressao-diferencial-mps20n0040d-d-4417.html> - Acessado em 17/05/2018.
- [12] Ferramenta para projeto de filtros Sallen-Key. Disponível em: <http://sim.okawa-denshi.jp/en/OPseikiHikeisan.html> - Acessado em 17/05/2018.
- [13] Nódulo sinoatrial e propagação do sinal cardíaco. Disponível em: <http://enfermagemitu.blogspot.com.br/2009/12/a-atividade-eletrica-do-coracao.html> - Acessado em 17/05/2018.
- [14] Pontos de contato dos eletrodos no corpo. Disponível em: <https://lifeinthefastlane.com/ecg-library/basics/lead-positioning/> - Acessado em 17/05/2018.
- [15] Módulo Wi-Fi ESP8266. Disponível em: <https://www.filipeflop.com/produto/modulo-wifi-esp8266-esp-01/#tab-blogrelacionados> - Acessado em 17/05/2018.
- [16] Especificações do Display. Disponível em: [https://www.ebay.com/itm/5-0-Polegadas-5-0-TFT-Lcd-Modulo-Display-SSD1963-com-painel-de-toque-cartao-SD-800X480-/201543907515?\\_trksid=p2349526.m4383.l4275.c10#viTabs\\_0](https://www.ebay.com/itm/5-0-Polegadas-5-0-TFT-Lcd-Modulo-Display-SSD1963-com-painel-de-toque-cartao-SD-800X480-/201543907515?_trksid=p2349526.m4383.l4275.c10#viTabs_0) - Acessado em 17/05/2018.
- [17] Shield regulador de tensão para o Display. Disponível em: [https://ae01.alicdn.com/kf/HTB1I41EaAfb\\_uJkSmLyq6AxoXXay/TFT-01-3-2-inch-LCD-Shield-Mega-Touch-Panel-Expansion-Board-Shield-v2-2-for-Arduino.jpg](https://ae01.alicdn.com/kf/HTB1I41EaAfb_uJkSmLyq6AxoXXay/TFT-01-3-2-inch-LCD-Shield-Mega-Touch-Panel-Expansion-Board-Shield-v2-2-for-Arduino.jpg) - Acessado em 17/05/2018.
- [18] Conexão entre Display, Shield e Arduino. Disponível em: <https://hifiduino.files.wordpress.com/2012/03/dsc01440.jpg> - Acessado em 17/05/2018.
- [19] Especificações do módulo carregador de bateria. Disponível em: <http://www.usinainfo.com.br/modulos-para-arduino/modulo-carregador-de-bateria-li-ion-tp4056->

2583.html?search\_query=%09Modulo+Carregador+de+Bateria+Li-Ion+-+TP4056&results=1 - Acessado em 17/05/2018.

[20] Imagem do circuito carregador de bateria. Disponível em: [http://img.dxcn.com/productimages/sku\\_373990\\_1.jpg](http://img.dxcn.com/productimages/sku_373990_1.jpg) - Acessado em 16/08/2017.

[21] Especificações do módulo regulador de tensão Step Up. Disponível em: <https://www.usinainfo.com.br/reguladores-de-tensao/regulador-de-tensao-ajustavel-mt3608-auto-boost-step-up-para-mais-25v-a-28v-4489.html#comentarios> - Acessado em 17/05/2018.

[22] Imagem do regulador de tensão. Disponível em: <https://www.addicore.com/MT3608-Boost-Converter-p/ad300.htm> - Acessado em 16/08/2017.

[23] LM7805. Disponível em: <https://www.sparkfun.com/datasheets/Components/LM7805.pdf> - Acessado em 11/08/2017.

[24] Caixa Patola. Disponível em: [https://www.eletopecas.com/\\_uploads/produtoArquivo/produtoArquivo\\_8766\\_1243\\_orig.jpg](https://www.eletopecas.com/_uploads/produtoArquivo/produtoArquivo_8766_1243_orig.jpg) - Acessado em 18/05/2018.

[25] SimBaby. Disponível em: <https://www.firehousemedical.com/store3/simbaby-main-product.html> - Acessado em 18/05/2018.

[27] Laerdal. Disponível em: <https://www.laerdal.com/images/L/ABOTTYDO.jpg> - Acessado em 18/05/2018.

[28] Monitor multiparamétrico. Disponível em: <http://www.cleanmedical.com.br/blog/monitores-multiparametros-o-que-e-11-5018-1044/> - Acessado em 11/08/2017.

[29] Datasheet do regulador de tensão. Disponível em: <https://www.olimex.com/Products/Breadboarding/BB-PWR-3608/resources/MT3608.pdf> - Acessado em 16/08/2017.

[30] Datasheet do circuito carregador. Disponível em:

<https://dlnmh9ip6v2uc.cloudfront.net/datasheets/Prototyping/TP4056.pdf> - Acessado em 16/08/2017.

[31] Regulador de tensão. Disponível em:  
<http://www.ti.com/lit/ds/symlink/lm317.pdf> - Acessado em 18/05/2018.







## APÊNDICE A – Código completo

```
#include <UTFT.h>
#include <URTouh.h>
UTFT myGLCD(TFT01_50,38,39,40,41);//
URTouh myTouch( 6, 5, 4, 3, 2);

// Armazena o estado dos botões
volatile bool valor_botao1 = 0;
volatile bool valor_botao2 = 0;
volatile bool valor_botao3 = 0;
volatile bool valor_botao4 = 0;
volatile bool valor_botao5 = 0;

//Configurações para o WIFI
String network = "Diego Roteador";
String password = "09031994";
#define IP "184.106.153.149"
String GET = "GET /update?key=4KRV2C2BLCU5ENEF";

extern uint8_t SmallFont[];// Declara a fonte que será utilizada
extern uint8_t BigFont[];// Declara a fonte que será utilizada
extern uint8_t SevenSegNumFont[];
extern unsigned int iconebateriafull[0x398];
extern unsigned int iconecoracao[0x400];
extern unsigned int iconeufsc[0x1E78];

int buf2[238];
int o=0;
int x=51;
//int w=152;
int w=51;
int z=253;
int y1;
int y2;
int y3;
int y4;
int a;
int valores[7];

int ym[]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
```

```
int yn[]={0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};
int y1p;
int y2p;

//oximetria
int analogPin=12;
float val=0;//int
float val1=0;//int
float R=0;
float Spo2=0;

int beat=0;
int bpm=0;

unsigned long previousMillis3 = 0;//spo2 sinal delay
unsigned long previousMillis5 = 0;//touch botão1
unsigned long previousMillis6 = 0;//touch botão2
//

void setup()
{
  Serial2.begin(115200);
  pinMode(8,OUTPUT);//RED
  pinMode(9,OUTPUT);//IR
  digitalWrite(8,LOW);
  digitalWrite(9,LOW);
  pinMode(11, OUTPUT);//pino 11 é a bomba
  pinMode(12, OUTPUT);//pino 12 é a válvula
  pinMode(19,INPUT);//era 21
  attachInterrupt(digitalPinToInterrupt(19), batimentos, CHANGE);//era
21
  pinMode(21,INPUT_PULLUP);//era 21
  attachInterrupt(digitalPinToInterrupt(21), botao1, FALLING);
  pinMode(20,INPUT_PULLUP);//era 21
  attachInterrupt(digitalPinToInterrupt(20), botao2, FALLING);
  pinMode(18,INPUT_PULLUP);//era 21
  attachInterrupt(digitalPinToInterrupt(18), botao3, FALLING);

  myGLCD.InitLCD();
  myGLCD.clrScr();
  myGLCD.setFont(BigFont);
```

```

myTouch.InitTouch();
myTouch.setPrecision(PREC_HI);
myGLCD.fillScr(VGA_BLACK);
myGLCD.drawBitmap(361, 80, 78, 100, iconeufsc);
myGLCD.setColor(255, 255, 255);
myGLCD.print("Monitor Multiparametrico: ", CENTER, 200);
myGLCD.print("Eletrocardiograma, Oximetria e Pressao Arterial.",
CENTER, 240);
myGLCD.print("Por: Diego Claudio da Silva.", CENTER, 280);
delay(7000);
myGLCD.fillScr(VGA_BLACK);
myGLCD.setFont(SmallFont);
myGLCD.print("Monitor Multiparametrico - Por: DCS", CENTER, 5);
//Desenha os retângulos dos botões e do gráfico
myGLCD.drawRoundRect(50, 30, 600, 479);//talvez 479
myGLCD.drawRoundRect(601, 30, 799, 479);
myGLCD.drawLine(615, 180, 785, 180);
myGLCD.drawLine(615, 330, 785, 330);

//Escreve os textos
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_GREEN);
myGLCD.print("ECG", 673, 35);
myGLCD.print("BPM", 673, 146);
myGLCD.setColor(VGA_AQUA);
myGLCD.print("SpO2", 665, 185);
myGLCD.print("%", 690, 290);
myGLCD.setColor(VGA_RED);
myGLCD.print("NiBP", 673, 335);
myGLCD.print("mmHg", 673, 445);
//botão enviar
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_GRAY);
myGLCD.fillRoundRect(0, 30, 49, 254);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 30, 49, 254);
myGLCD.setFont(BigFont);
myGLCD.setBackColor(VGA_TRANSPARENT);
myGLCD.print("E", 17, 94);
myGLCD.print("n", 17, 110);
myGLCD.print("v", 17, 126);

```

```

myGLCD.print("i", 17, 142);
myGLCD.print("a", 17, 158);
myGLCD.print("r", 17, 174);
//botão clear
myGLCD.setColor(VGA_GRAY);
myGLCD.fillRoundRect(0, 255, 49, 479);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 255, 49, 479);
myGLCD.setFont(BigFont);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print("L", 17, 318);
myGLCD.print("i", 17, 334);
myGLCD.print("m", 17, 350);
myGLCD.print("p", 17, 366);
myGLCD.print("a", 17, 382);
myGLCD.print("r", 17, 398);

myGLCD.drawBitmap(750, 5, 46, 20, iconebateriafull);
}

void loop()
{
  touch();
  if(valor_botao1 == 1)
  {
    bpmecg();
    printsinal();
    touch();
  }

  if(valor_botao2 == 1)
  {
    printsinal1();
    oximetria();
    touch();
  }

  if(valor_botao3 == 1)
  {
    pressao();
    delay(3000);
  }
}

```

```

valor_botao3 = 0;
touch();
}

while(valor_botao4 == 1)
{
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_TEAL);
myGLCD.fillRoundRect(0, 30, 49, 254);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 30, 49, 254);
myGLCD.setFont(BigFont);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print("E", 17, 78);
myGLCD.print("n", 17, 94);
myGLCD.print("v", 17, 110);
myGLCD.print("i", 17, 126);
myGLCD.print("a", 17, 142);
myGLCD.print("n", 17, 158);
myGLCD.print("d", 17, 174);
myGLCD.print("o", 17, 190);
setupEsp8266();
updateDados(String(valores[4])
,String(valores[1]),String(valores[2]),String(valores[3]));
delay(30000);
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_GRAY);
myGLCD.fillRoundRect(0, 30, 49, 254);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 30, 49, 254);
myGLCD.setFont(BigFont);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print("E", 17, 94);
myGLCD.print("n", 17, 110);
myGLCD.print("v", 17, 126);
myGLCD.print("i", 17, 142);
myGLCD.print("a", 17, 158);
myGLCD.print("r", 17, 174);
valor_botao4=0;
}

```

```

while(valor_botao5 == 1)
{
myGLCD.setFont(BigFont);
myGLCD.setColor(VGA_TEAL);
myGLCD.fillRoundRect(0, 255, 49, 479);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 255, 49, 479);
myGLCD.setFont(BigFont);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print("L", 17, 303);
myGLCD.print("i", 17, 319);
myGLCD.print("m", 17, 335);
myGLCD.print("p", 17, 351);
myGLCD.print("a", 17, 367);
myGLCD.print("n", 17, 383);
myGLCD.print("d", 17, 399);
myGLCD.print("o", 17, 415);

myGLCD.setColor(VGA_BLACK);
myGLCD.fillRoundRect(51, 31, 599, 478);//talvez 479
myGLCD.fillRoundRect(602, 52, 798, 145);
myGLCD.fillRoundRect(602, 202, 798, 289);
myGLCD.fillRoundRect(602, 352, 798, 444);

myGLCD.setColor(VGA_GRAY);
myGLCD.fillRoundRect(0, 255, 49, 479);
myGLCD.setColor(VGA_WHITE);
myGLCD.drawRoundRect(0, 255, 49, 479);
myGLCD.setFont(BigFont);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.print("L", 17, 318);
myGLCD.print("i", 17, 334);
myGLCD.print("m", 17, 350);
myGLCD.print("p", 17, 366);
myGLCD.print("a", 17, 382);
myGLCD.print("r", 17, 398);
valor_botao5=0;
}
}

void touch(){

```



```

//Configurações para o touch
long x, y;

if (myTouch.dataAvailable() == true)
{
    myTouch.read();
    x = myTouch.getX();
    y = myTouch.getY();

    if ((x > 600) && (x < 799))

    {
        if ((y > 30) && (y < 180))
        {
            unsigned long currentMillis5 = millis();

            if (currentMillis5 - previousMillis5 >= 700) {
                previousMillis5 = currentMillis5;

                if (valor_botao1 == 0) {
                    valor_botao1 = 1;
                } else {
                    valor_botao1 = 0;
                }
            }
        }

        if ((y > 181) && (y < 330))
        {
            unsigned long currentMillis6 = millis();

            if (currentMillis6 - previousMillis6 >= 700) {
                previousMillis6 = currentMillis6;

                if (valor_botao2 == 0) {
                    valor_botao2 = 1;
                    digitalWrite(8,HIGH);
                } else {
                    valor_botao2 = 0;
                    digitalWrite(8,LOW);
                }
            }
        }
    }
}

```

```

    }
}
    if ((y>331) && (y<479))
{
    if(valor_botao3 == 0){
        valor_botao3 = 1;
    }
    else
    {
        valor_botao3 = 0;
    }
}
}
    if ((x > 0) && (x<49))

{
    if ((y>30) && (y<254))
    {
        if(valor_botao4 == 0){
            valor_botao4 = 1;
        }
        else
        {
            valor_botao4 = 0;
        }
    }

    if ((y>254) && (y<479))
    {
        if(valor_botao5 == 0){
            valor_botao5 = 1;
        }
        else
        {
            valor_botao5 = 0;
        }
    }
}
}
}

```

```

void setupEsp8266()
{
    Serial2.flush();
    Serial2.println(F("AT+RST"));
    delay(7000);
    if (Serial2.find("OK"))
    {
        Serial2.flush();
        changingMode();
        delay(5000);
        Serial2.flush();
        connectToWiFi();
    }
    else
    {
    }
}

//-----
// sets esp8266 to station mode
//-----
bool changingMode()
{
    Serial2.println(F("AT+CWMODE=1"));
    if (Serial2.find("OK"))
    {
        return true;
    }
    else if(Serial2.find("NO CHANGE")){
        return true;
    }
    else
    {
        return false;
    }
}

//-----
// connect esp8266 to wifi access point
//-----

```

```

bool connectToWiFi()
{
  String cmd = F("AT+CWJAP=\"");
  cmd += network;
  cmd += F("\",\"");
  cmd += password;
  cmd += F("\"");
  Serial2.println(cmd);
  delay(15000);

  if (Serial2.find("OK"))
  {
    return true;
  }
  else
  {
    return false;
  }
}

//-----
// sends sensor data to thingspeak.com
//-----
void updateDatos(String ECG,String SpO2,String PA,String Temp)
{
  String cmd = "AT+CIPSTART=\"TCP\", \"";
  cmd += IP;
  cmd += "\",80";
  Serial2.println(cmd);
  delay(5000);
  if(Serial2.find("Error")){
    return;
  }
  cmd = GET + "&field1=" + ECG + "&field2=" + SpO2 + "&field3=" +
PA + "&field4=" + Temp + "\r\n";
  Serial2.print("AT+CIPSEND=");
  Serial2.println(cmd.length());
  delay(15000);
  if(Serial2.find(">"))
  {
    Serial2.print(cmd);
  }
}

```

```

    }else
    {
        Serial2.println("AT+CIPCLOSE");
    }
}

//Função para o calculo da concentração de SpO2
void oximetria() {
    if(analogRead(12)==LOW){
        val1=analogRead(13);
    }
    if(analogRead(12)>500){
        val=analogRead(13);
    }
    R=100*(val/val1);
    Spo2=-25*R;
    Spo2=Spo2+110;

    if (R>100 && R<251){
        valores[1]=100-25*log10(R/100);
    }
}

void pressao() {

    myGLCD.setColor(VGA_BLACK);
    myGLCD.fillRoundRect(602, 364, 798, 444);
    myGLCD.setFont(BigFont);
    myGLCD.setColor(VGA_RED);
    myGLCD.setBackgroundColor(VGA_TRANSPARENT);
    myGLCD.print("Aguarde", 645, 365);

    digitalWrite(11, HIGH);
    digitalWrite(12, HIGH);
    int vsensor=analogRead(A10);
    while (vsensor<750){
        vsensor=analogRead(A10);
        delay(5000);
        touch();
    }
}

```

[illegible]

```

myGLCD.fillRect(602, 364, 798, 444);
myGLCD.setFont(SevenSegNumFont);
myGLCD.setColor(VGA_RED);
myGLCD.setBackgroundColor(VGA_TRANSPARENT);
myGLCD.printNumI(SYS, 602, 365);
myGLCD.drawLine(700, 410, 705, 370);
myGLCD.drawLine(701, 410, 706, 370);
myGLCD.drawLine(702, 410, 707, 370);
myGLCD.drawLine(703, 410, 708, 370);
myGLCD.drawLine(704, 410, 709, 370);
myGLCD.printNumI(DIA, 710, 365);
myGLCD.setFont(BigFont);
myGLCD.printNumI(MAP, 684, 420);
tone(10,294,300);
delay(1000);
tone(10,294,300);
delay(1000);
}

```

//Função para desenhar os gráficos na tela.

```
void printsinal() {
```

```

    x++;
    if (x==599){
        x=52;//ERA 51
        myGLCD.setColor(0,0,0);
        myGLCD.fillRect(51, 32, 599, 255);
        myGLCD.fillRect(0, 0, 80, 28);
        myGLCD.fillRect(620, 54, 797, 140);
        myGLCD.setColor(VGA_GREEN);
        myGLCD.printNumI(valores[4], 665, 74);
    }

```

```

    for (int i=0;i<20;i=i+1){
        int sensorValue = analogRead(A9);
        yn[i] = 255-44*(sensorValue * (5.0 / 1023.0));
    }

```

```

y2=(yn[0]+yn[1]+yn[2]+yn[3]+yn[4]+yn[5]+yn[6]+yn[7]+yn[8]+yn[9]
+yn[10]+yn[11]+yn[12]+yn[13]+yn[14]+yn[15]+yn[16]+yn[17]+yn[18]
+yn[19])/20;
myGLCD.setColor(0,255,0);

```

```

    myGLCD.drawLine((x-1),y2p,x,y2);
    y2p=y2;
}

void printsinal1() {
    w++;
    if (w==599){
        w=52;
        myGLCD.setColor(0,0,0);
        myGLCD.fillRect(51, 256, 599, 478);
        myGLCD.fillRect(0, 0, 80, 28);
        myGLCD.fillRect(620, 221, 797, 276);
        myGLCD.setColor(0, 255, 255);
        myGLCD.setFont(SevenSegNumFont);
        myGLCD.printNumI(valores[1], 665, 221);
    }

    for (int i=0;i<20;i=i+1){
        int sensorValue = analogRead(A13);
        ym[i] = 478-44*(sensorValue * (5.0 / 1023.0));
    }

    y1=(ym[0]+ym[1]+ym[2]+ym[3]+ym[4]+ym[5]+ym[6]+ym[7]+ym[8]+
ym[9]+ym[10]+ym[11]+ym[12]+ym[13]+ym[14]+ym[15]+ym[16]+ym
[17]+ym[18]+ym[19])/20;
    myGLCD.setColor(0,255,255);
    myGLCD.drawLine((w-1),y1p,w,y1);
    y1p=y1;
}

void bpmecg(){
    unsigned long currentMillis3 = millis();

    if (currentMillis3 - previousMillis3 >= 10000) {
        bpm=(beat*60*0.028)/10;
        valores[4]=bpm;
        beat=0;
        previousMillis3 = currentMillis3;
    }
}

```



```

void batimentos(){
if (analogRead(A9) < 450){
beat++;
}
}
void botao1(){
    if (valor_botao1) {
        valor_botao1 = false;
        //tone(10,392,200);
        digitalWrite(8,LOW);
        tone(10,392,200);
        delay(1000);
    } else {
        valor_botao1 = true;
        digitalWrite(8,LOW);
        tone(10,392,200);
        delay(1000);
    }
}

void botao2(){
    if (valor_botao2) {
        valor_botao1 = false;
        valor_botao2 = false;
        digitalWrite(8,LOW);
        tone(10,392,200);
        delay(1000);
    } else {
        valor_botao1 = true;
        valor_botao2 = true;
        digitalWrite(8,HIGH);
        tone(10,392,200);
        delay(1000);
    }
}

void botao3(){
    if (valor_botao3) {
        valor_botao3 = false;
        tone(10,392,200);
        delay(1000);
    } else {

```

```
        valor_botao3 = true;
        tone(10,392,200);
        delay(1000);
    }
}
```